Privacy - Preserving Data Exchange and Aggregation in Healthcare

THÈSE N^{O} 8310 (2018)

PRÉSENTÉE LE 22 AOÛT 2018

À LA FACULTÉ INFORMATIQUE ET COMMUNICATIONS
LABORATOIRE DE SYSTÈMES D'INFORMATION RÉPARTIS
PROGRAMME DOCTORAL EN INFORMATIQUE ET COMMUNICATIONS

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Alevtina DUBOVITSKAYA

acceptée sur proposition du jury:

Prof. R. Urbanke, président du jury
Prof. K. Aberer, Prof. M. I. Schumacher, directeurs de thèse
Prof. K. Huguenin, rapporteur
Prof. F. Wang, rapporteur
Prof. B. A. Ford, rapporteur

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE
Suisse
2018

To my family

Success is not final, failure is not fatal:
it is the courage to continue that counts.
– *Winston Churchill*

Acknowledgements

I would like to thank my co-supervisors, Prof. Michael Schumacher and Prof. Karl Aberer, for their support and guidance. Michael, thank you for having faith in me, for giving me the opportunity to work on such an interesting and exciting ISyPeM2 project and to gain a valuable experience visiting research groups abroad. Karl, thank you for helping me with the most difficult part, improving and finalizing my thesis. I would like to thank the jury members, Prof. Bryan Ford, Prof. Kévin Huguenin, Prof. Fusheng Wang, and the jury president Prof. Rüdiger Urbanke, for their valuable constructive comments, for their support, and for the time they took to help me to improve my thesis. Kévin, thank you for your guidance, especially during my first and last years of studies.

I would like to thank my co-authors for their collaboration and their contributions to this thesis, I learned a lot from them: Prof. Visara Urovi, Dr. Matteo Vasirani, Prof. Yann Thoma, Prof. Thierry Buclin, Prof. Fusheng Wang, Prof. Zhigang Xu, Dr. Davide Calvaresi, Dr. Jean-Paul Calbimonte, and Imanol Barba. I also would like to thank Prof. Josep Domingo-Ferrer, Prof. Jean-Pierre Hubaux, Prof. Phillip Janson, Prof. Josep Pegueroles, and Prof. Fusheng Wang for the opportunity to work and to discuss my work with them. I am grateful to Nano-Tera for funding the ISyPeM2 project, and to the members of the teams I had a chance to work with in the framework of ISyPeM2: REDS @HEIG, department of Clinical Pharmacology and Laboratories department @CHUV, research groups @EPFL and @HES-SO.

I would like to express my gratitude to Chantal Francois, Margaret Escandari, Anne Darbellay, and Claudia Mathieu for their help and support solving different administrative issues. I would like to thank Holly Cogliati for the careful editing and proof-reading of my thesis and for the patience she had writing “the” for the hundred-and-first time.

I thank my friends and colleagues who I met at EPFL: Berker, Brunella, Giel, Jean-Eudes, Julia, Lucas, Mehdi, Mohammad, Panayotis, Régis, and Sonia. I thank my friends and colleagues I met at HES-SO: Alba, Albert, Ale, Antonio, Davide, Dmitri, Fabian, Fabien, Francesca, François, Gaetano, Guillaume, Guiseppe, Ivan, Jérôme, Manfredo, Mara, Matthieu, Matteo, Mariam, Meri, Michael, Morgane, Nancy, Oscar, Paola, Paul, Ranveer, Roger, Sebastian, Stefan, Stefano B., Stefano M., Thomas, Valerio, Vincent, and Yashin. I would like to thank you all for your friendship, support, help, sense of humour, patience, and for filling the PhD journey with emotions, crazy coffee breaks, hikes, climbing, ping-pong, badminton, billiard, salsa, violin, and apéros.

Acknowledgements

Brunella, thank you for simply always being there during all these years from the very first days we started at EPFL. Davide, thank you for your contagious research enthusiasm and for the enjoyment to work with you. Fabien, thank you for always spotting the funny part in all my complains about life. Gaetano, thank you for all the discussions that we had and for sharing with me my affection to "the philosophy of Socrates". Jean-Eudes, thank you for your support and for the coffee breaks. Julia, thank you for always being ready to listen and to find a getaway from any panic situations. Thomas, thank you for being so easy to talk to and for your advice.

During these years in Switzerland I was very lucky to have friends outside work as well: Boris, Delphine, Jérémie, Karina, Katya, Marina S., Marina Sh., Sofia, Vanya, Yuan, and Ziggy. I also truly enjoyed the time I spent in Barcelona with Ana, Mladen, Nikola, and Suzana.

I would like to thank my friends from Russia, for whom the almost 3000 km to Switzerland were the same as 30 km to Domodedovo, who came to visit me here and, especially during my first years, showed me how beautiful Switzerland is, and who are still planning to visit: Alevtina, Alisa, Anya, Dasha, Den, Denis Z., Lena, Misha, Nastya, Nikita, and Stas. Moroz, special thanks go to you for always being there to help me and to share with me whatever happens, regardless of the distance, the time of a day (or a night), and the size of the problem or the gravity of the news.

My warmest gratitude goes to my family and in-laws for their love and care. I thank you for showing me by your example how to work hard and overcome difficulties, for letting me know that I always have your support, and for helping and encouraging me in everything I do. Masha, thank you for being my wise sister, for dragging me into the computer science world, and for setting the bar (very) high. Luc, I would like to thank you for telling me once back in 2011 about EPFL, and not only for encouraging me to apply for the PhD program, but simply being there, even for the multiple rehearsals of my candidacy exam and my thesis defense. Mama and Papa, thank you for your unconditional love and patience, for helping me not to be afraid of difficulties, but see them as an experience that makes me stronger, and for teaching me that no matter what I am working on, I should always do my best. Damien, thank you for teaching me how to bike, ski, and to be more organized (and for being so patient) and for the joy you bring in my life every day.

Abstract

Medical data are often scattered among multiple clinics, hospitals, insurance companies, pharmacies, and research institutions that store and process personal healthcare information. The use of information and communication technologies for health (eHealth) provides us with the means to share healthcare data between authorized parties in an efficient manner.
In this thesis, we address some of the challenges of implementing eHealth in practice: to achieve interoperability between data sources, and to ensure privacy for patients. Achieving both of these guarantees is our goal but they seem conflictual, hence the challenge. Once interoperability is achieved and a patient's data are shared, it becomes even more difficult to ensure the patient's privacy i.e., to provide to a patient control over his data and to guarantee the data anonymity in medical research. We address the aforementioned challenges by studying requirements from medical and legal perspectives, and by developing algorithms and frameworks to support privacy-preserving dynamic data-sharing, exchange, and aggregation from multiple data sources.
In the first part of the thesis, we address certain privacy challenges. We present a framework based on the blockchain technology for ensuring traceability and accountability when sharing, exchanging, and aggregating medical data. Our framework ensures privacy, security, availability, and fine-grained access control over highly sensitive patient-data. We also analyze the potential of applying blockchain technology in different eHealth settings: primary care, medical-data research, and connected health. Our second contribution is a framework for privacy-preserving data aggregation: an algorithm for constructing the anonymized database and a protocol that improves the utility of the anonymized database as the database grows.
In the second part of the thesis, we focus on achieving interoperability. We design an interface specification that defines communication protocols and messages supporting integration of a new software tool in clinical practice. Then, we develop a multi-agent system (MAS) for the dynamic aggregation of the data collected and generated by this software tool for the purpose of clinical research. This MAS takes into account the objectives of the research study, the availability of data, and could employ our proposed algorithm for privacy-preserving data aggregation. The negotiation protocol in the framework of the MAS achieves a precise definition of database characteristics, such as schema, content, and privacy parameters, therefore increasing the efficiency of data collection for medical research and ensuring the privacy of patients.

Key words: eHeatlh, Privacy, Interoperability, Blockchain, Medical Data Sharing, Medical Data Exchange, Medical Data Aggregation, De-generalization of Anonymized Data, Data Utility.

Résumé

Les données médicales sont souvent dispersées entre les cliniques, les hôpitaux, les compagnies d'assurance, les pharmacies, et les instituts de recherche qui collectent et traitent des données relatif aux services de santé. Les progrès informatique et la digitalisation des données médicales (l'eHealth) offrent la possibilité de partager les données de santé entre différents acteurs autorisés d'une manière efficace.

Dans cette thèse, nous adressons certains défis liés à la mise en oeuvre de l'eHealth en pratique. Dans un premier temps en accomplissant l'interopérabilité entre différentes sources de données, et dans un second temps en assurant la confidentialité des données patient. Ces défis intéragissent et peuvent entrer en conflit. Une fois l'interopérabilité accomplie et les données patient partagées, il devient encore plus difficile d'assurer la confidentialité, c'est à dire d'offrir la possibilité au patient de contrôler les accès à ses données et de garantir son anonymat lors de recherches médicales scientifique. Nous approchons ces défis en étudiant les besoins médicaux et légaux, et en développant des algorithmes et systèmes qui soutiennent un partage de données dynamique et confidentiel dans un cadre d'échanges entre différentes sources.

La première partie de cette thèse répond au défi de confidentialité. Nous présentons un système basé sur la technologie blockchain qui assure la traçabilité et donc la responsabilité lors du partage, des échanges, et de la collecte de données médicales. Le système proposé assure la condifidentialité, la securité, la disponibilité, et un contrôle précis des données sensibles patient. Nous analysons également les bénéficent potentiels à l'application de la blockchain pour différentes problématiques de l'eHealth : les soins médicaux, les données médicales pour la recherche, et la santé connectée. Notre seconde contribution est un système pour l'agrégation de données assurant la confidentialité, c'est à dire un algorithme pour construire des bases de données anonymisées pour la recherche ainsi qu'un protocole qui améliore l'utilité de ces bases de données à mesure que leurs volumes augmentent.

La seconde partie de cette thèse répond au défi d'interopérabilité. Nous concevons une interface qui définit les protocoles de communication et les formats de messages supportant l'intégration de nouveaux logiciels informatique pour la pratique médicale. Ensuite, nous développons un système multi agents (SMA) d'agrégation dynamique de données recueillis et générées par cet outil pour la recherche médicale. Le SMA prend en compte les objectifs de la recherche, la disponibilité des données, et peut être combiné avec notre algorithme d'anonymisation de données. Le protocole de négociation dans le système SMA vise à définir de façon précise les caractéristiques de la base données demandée tel que le schéma, les

champs, et les paramètres de confidentialité, de façon à augmenter l'efficacité de la collecte des données pour la recherche tout en assurant la confidentialité des patients.

Mots clefs : eHeatlh, Confidentialité, Interopérabilité, Blockchain, Partage et échange de données médicales, Agrégation de données médicales, Déanonymisation de données confidentielles, Utilité de données médicales.

Contents

List of Figures

List of Tables

1 Introduction

A person's health-related information evolves over time and can be stored and processed by multiple entities in various formats and volumes, and for different purposes. A patient might visit several hospitals during treatment or rehabilitation. Therefore, the patient's data are scattered among numerous clinics, hospitals, medical devices, and laboratories. Healthcare data are also used in medical research to build mathematical models for personalized medicine, in clinical trials, and during the process of drug development. Retrospective medical-data are employed for comparative effectiveness research and in public health.

The use of information and communication technologies (ICT) for health - eHealth - aims at providing the possibility to share and exchange, in an efficient manner, healthcare data among authorized parties. One of the main concepts in eHealth is electronic health records (EHRs). EHRs are computerized medical information systems that collect, store, and display patient information [ABT13]. EHRs contain lab tests, prescriptions, images, demography, history, payment information, etc. The ability to access data from EHRs is beneficial for all participants of a healthcare system. Having access to a detailed patient record helps healthcare experts to make better treatment decisions, reduces the cost of providing ambulatory care, enables the collection of high quality data in a short period of time for research purposes, and simplifies the data-management process for patients, especially in case of a chronic disease.

The use of EHRs is being adopted all over the world. According to the World Health Organization[1], 27 out of 53 countries (59%) of the European Region have a national electronic health record system. In the United States, The Health Information Technology for Economic and Clinical Health (HITECH) Act provided a series of incentives to encourage a widespread EHR adoption in hospitals and among office-based physicians. However, despite the efforts to put eHealth in practice and the positive effects of the EHRs usage in medical practices, the adoption of such systems meets resistance due to the existing barriers to implementing EHRs [ABT13]. Many studies [GPW+14, CMGP10, ABT13, SKC+07, BB10, O+16] reported interoperability as an important barrier to implementing EHRs. For example, the hardware and

[1] http://www.who.int

software for EHRs cannot be used "straight out of the box", it requires interconnectivity with other devices that "complement" the EHR system [BB10].

Interoperability is the ability of systems and devices to exchange and interpret data. Achieving interoperability will certainly advance the adoption of eHealth. However, when managing highly sensitive information such as health-related data, interoperability could lead to violations of the patients' privacy. The consequences could be irreversible and have a long-term impact on the patient, as well as on his social environment.

From a philosophical point of view, [Sch84, Wal02] privacy can be defined along three main lines. First, privacy as a person's will to determine *which personal information* may be communicated to others, second, privacy in relation to a person's *control* over access to the information about himself, and, third, privacy as *limited access* to the personal information. Therefore, we could define a patient's privacy as his right to trace and control the personal data that flow between various systems and are shared among different peers. More specifically, in the framework of collecting anonymized data for medical research, a patient's right to privacy is the right to remain anonymous. According to the existing regulations, every time the data are shared, the patient has to provide a signed consent that specifies the terms of data sharing (an access control policy). However, consents are still paper based and are not sufficient, which means that it is very difficult for the patient to express his specific access control policies.

If the data are anonymized such that re-identification of a person is not reasonably possible, no consent is required[2]. However, anonymization can limits the quality of the data [RSH07, BS08, LL09], hence their utility for research. Moreover, when aggregating information about a person from multiple sources, the risk of re-identification increases, hence raising a privacy concern. Additionally, a further increase of the data-anonymity level reduces the data utility even further. If the data source is large enough, the relevant information can be preserved even after applying anonymization. However, constructing the sufficiently large database can require significant amount of time and can delay the research process. Therefore, the mechanisms for privacy-preserving utility-aware dynamic data aggregation are required to provide the appropriate privacy levels and to support medical research.

1.1 Research Questions and Contributions

In this work, we strive to develop the best approaches to achieve interoperability, support dynamicity in eHealth, and to ensure patients' privacy. We define the research questions (*RQ#*) and describe our contributions below.

- *RQ1*: How do we set up and guarantee traceability, accountability and fine-grained access control over healthcare data?

  Contribution: To address this question, we analyze the existing blockchain technology

[2] EC Data Protection Directive 95/46/EC

implementations and the possibility of using them in different eHeath scenarios: *(i)* sharing data during treatment and post-treatment monitoring; *(ii)* aggregating data for the purposes of medical research; and *(iii)* exchanging data in the context of connected health[3]. We show how limitations of the existing healthcare systems could be overcome by employing blockchain technology in healthcare. We present a framework for blockchain-based data sharing for the primary care of oncology patients under anti-cancer treatment. In collaboration with the Department of Radiation Oncology of a major US hospital, we developed a prototype of this framework. The functionality of the prototype meets the requirements from a medical practice perspective. The prototype ensures privacy, security, availability, and fine-grained access control over the EHR data. We also proposed an extension of the current prototype to enable patients to provide their consent for data aggregation for research purposes. Our framework for EHR-data sharing that uses permissionned blockchain technology is the first work in this field. Once adopted by the health community, it will reduce the turnaround time for EHR-data sharing, improve decision making for medical care, and reduce overall costs.

- *RQ2*: How do we preserve the patient's privacy and improve data utility when aggregating locally anonymized datasets that could contain data about the same patient?

  Contribution: We design and evaluate an algorithm for dynamic and distributed privacy-preserving data aggregation. We use pseudonymization and anonymization techniques to satisfy the k-anonymity requirement that is common for medical research. To mitigate the problem imposed by data anonymization, specifically, limited data quality, hence limited utility of the data aggregated for research purposes, we propose to update in a privacy-preserving way the resulting anonymized database with the de-generalized data from the initial sources. To achieve this, we study and formalize the data de-generalization criteria in a distributed environment. Then, we implement and evaluate a one-step de-generalization protocol that enables us to improve the utility of the data as the database grows. In order to de-generalize the data while preserving the k-anonymity level, we employ efficient functional encryption and secret-sharing schemes.

- *RQ3*: How do we ensure interoperability in order to seamlessly integrate a software tool into the existing network of electronic medical records, laboratory information-systems and other medical applications? How do we address issues such as different interfaces, data formats, and comprehensive clinical dataflow?

  Contribution: We address this question in the context of therapeutic drug monitoring (TDM), a key concept in precision medicine. The purpose of TDM is to avoid therapeutic failures or toxic effects of a drug due to the insufficient or excessive circulating concentration related to the variability between patients. We focus on the process of integrating *TUCUXI*: – a software tool for automatization of TDM – in a clinical workflow. We first study and redesign the existing data-flow in a university hospital. Then, using

[3]Connected health is a model for healthcare delivery that aims to maximize healthcare resources and provide opportunities for consumers to engage with caregivers and improve self-management of a health condition.

the functionality of the software, we design and implement the interfaces and message flows to achieve interoperability with the clinical database-management system and a successful integration into clinical practice. We also discuss the ethical issues related to the use of an automated decision-support system in clinical practice, in particular if it permits data aggregation for research purposes.

- *RQ4*: How do we improve the efficiency of the data-aggregation process according to the requirements of a research study?

  Contribution: The collection of medical data for research purposes is a challenging and long-lasting process, hence we need to accelerate and facilitate this process. We propose a new framework for the dynamic aggregation of the medical data from distributed sources. We create an agent-based coordination framework to be used between medical and research institutions. Our system employs principles of peer-to-peer network organization and coordination models *(i)* to search over existing distributed databases, and *(ii)* when a new database is needed, to identify the potential contributors. Our framework takes into account both the requirements of a research study and the current data availability, which leads to a better definition of database characteristics such as the schema, content, and privacy parameters. The algorithm for data aggregation presented in this thesis could be employed in this framework to ensure privacy-preserving interoperability for data aggregation in medical research.

1.2 Thesis Outline

The rest of the thesis is organized as follows. In Chapter 2.1, we present the specifics of healthcare-data management and a practical scenario employed in our work. We also list the international regulations that have to be applied when processing medical data. Then, we review the existing approaches for ensuring interoperability and privacy in healthcare scenarios, and we underline their limitations.

We devote Part 1 to ensuring privacy in two contexts. First, in Chapter 2.1, for the patient to keep control over data and to reinforce the desired access control policy, we present a new framework that is based on the blockchain technology for providing traceability of data. The content of the chapter is based on the following peer-reviewed publications:

- [DXR+17c] Alevtina Dubovitskaya, Zhigang Xu, Samuel Ryu, Michael Schumacher and Fusheng Wang: Secure and Trustable Electronic Medical Records Sharing using Blockchain. In *AMIA Annual Symposium Proceedings*, volume 2017, page 650, American Medical Informatics Association, 2017

- [DXR+17b] Alevtina Dubovitskaya, Zhigang Xu, Samuel Ryu, Michael Ignaz Schumacher and Fusheng Wang: How Blockchain could Empower eHealth: an Application for Radiation Oncology. In *VLDB Workshop on Data Management and Analytics for Medicine and*

Healthcare, pages 3-6, Springer, 2017.

- [DXR+17a] Alevtina Dubovitskaya, Zhigang Xu, Samuel Ryu, Michael Schumacher and Fusheng Wang, Blockchain dans la eSanté: Perspectives et une Application pour le Traitement Quotidien. *Swiss Medical Informatics*, Vol. 33, 2017.

In chapter 4, we describe an algorithm for efficient healthcare-data aggregation from multiple sources, and a distributed privacy-preserving protocol for improving, as the database grows, the utility of anonymized data. The content of the chapter is based on the following two peer-reviewed publications and one article in preparation:

- [DUV+15] Alevtina Dubovitskaya, Visara Urovi, Matteo Vasirani, Karl Aberer, and Michael I. Schumacher. A cloud-based eHealth architecture for privacy-preserving data-integration. In *IFIP International Information Security Conference*, pages 585-598. Springer,2015.
- [DUV+14] Alevtina Valeryevna Dubovitskaya, Visara Urovi, Matteo Vasirani, Karl Aberer, Aline Fuchs, Thierry Buclin, Yann Thoma, Michael I. Schumacher: Privacy-preserving interoperability for personalized medicine. *Swiss Medical Informatics*, Vol. 30, 2014.
- Alevtina Dubovitskaya, Michael Schumacher, and Karl Aberer. Improving the Utility of Incremental k-anonymous Datasets in Distributed Settings (submitted).

In Part 2 of the thesis, we focus on achieving interoperability, in particular for primary care (Chapter 5) and research purposes (Chapter 6). In Chapter 5, we describe the system developed for automatization of the TDM process and its integration into clinical dataflow. The content of this chapter is based on the following peer-reviewed publication:

- [DBS+17] Alevtina Dubovitskaya, Thierry Buclin, Michael Schumacher, Karl Aberer, and Yann Thoma. 2017. TUCUXI: An Intelligent System for Personalized Medicine from Individualization of Treatments to Research Databases and Back. In *Proceedings of the 8th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*, ACM-BCB '17, pages 223-232, New York, NY, USA, 2017.

In Chapter 6, we present the framework for dynamic aggregation of medical data from distributed sources. The content is based on the following peer-reviewed journal publication:

- [DUB+16] Alevtina Dubovitskaya, Visara Urovi, Imanol Barba, Karl Aberer, and Michael Ignaz Schumacher. A Multiagent System for Dynamic Data Aggregation in Medical Research. *BioMed research international*, 2016.

In Chapter 7, we conclude this thesis and provide a discussion about the potential applications of this work in other fields, as well as possible future research directions.

Background Part

2 Context and Technical Background

Achieving interoperability and privacy in healthcare settings requires an interdisciplinary approach to system design. First, it is important to understand the context of healthcare data management: what are the needs of the different actors in the healthcare environment, how is it regulated, and what are the desired functionalities and properties of the system. The next step is to analyze existing solutions, their limitations, as well as the technologies and primitives that could be used as building blocks to partially or completely satisfy the specified requirements. This chapter first describes the specifics of medical data management. Then, we review existing cryptographic primitives and data privacy techniques employed in this thesis for sharing, exchanging and aggregating medical data. Last, we provide an overview of related work together with an emphasis on some limitations that we address in the next chapters.

2.1 Healthcare Data Management

This section first introduces the context in which we address the research questions, in particular we describe two use cases. Then, we summarize the existing requirements and regulations that govern healthcare data management. Based on the context and existing regulations we then list security and privacy properties for the systems developed and presented in the next chapters.

2.1.1 Use case scenarios

In this Section we present the two main use cases of this thesis.

Treatment and Monitoring of Oncology Patients
A patient with a chronic disease or a serious medical condition may need to keep track of his medical record through his life, or may need to delegate the data management to his relatives. Management of medical history, access control, prescriptions, medical expenses, insurance correspondence, and payments is unavoidable, notoriously time-consuming, and bothersome

for most of these patients. In particular, for oncology patients, diagnosis and treatment at multiple hospitals are common. During the anti-cancer treatment or the post-treatment monitoring, it may be urgently required to know the radiation dose received during the patent's life-long treatment. Therefore, it is crucial for the patient to maintain his medical history and to be able to access or share his medical data during the treatment and post-treatment monitoring. Consent management procedures and data transfer may delay a critical treatment. Moreover, the inability to access complete medical history of the patient may result in harmful consequences for the patient. How can we guarantee that the patient's data are complete, stored securely, and can be accessed according to the patient consent in a fast and convenient manner?

Currently, if any of these data have to be transferred from Hospital 1 to Hospital 2, the following procedure takes place. First, the patient (or his official representative) has to sign a consent form – a document that specifies the data to be transferred and contains the information about the recipient of the data (Hospital 2). Then, the information has to be printed and mailed to the recipient. Consent management and data transfer in this case can become complicated and inconvenient: the patient may need to contact the caregiver and sign a consent form in the hospital where he is not receiving care anymore. Data transfer can take time, and on receiving the hard copy of the patient data, a clinician will have to introduce them into the system again. Moreover, with this approach, it is very difficult for the patient to know where the data are stored and to maintain access-control policies.

Oncology information systems are widely used to facilitate oncology specific comprehensive information and images management. For instance, the Aria medical system (Varian Medical Systems, Inc., Palo Alto, CA) combines radiation, medical and surgical oncology information, and can assist clinicians to manage different kinds of medical data, develop oncology-specific care plans, and monitor the radiation dose of patients. Yet, issues related to consent management, in particular in decentralized settings, are not addressed by this system.

In collaboration with the department of radiation oncology at Stony Brook University Hospital in the United States, we aim to provide traceability and accountability of the shared healthcare data and to simplify the process of decentralized consent management, for instance to be able to deploy our solution on top of the oncology-specific database-management system to address the data management issues listed above. The framework is presented and discussed in Chapter 3.

Therapeutic Drug Monitoring

The treatment of certain diseases, such as cancer, HIV, or other serious medical conditions, crucially relies on the administration of the drugs required to keep such life-threatening diseases under control. These drugs (e.g., *Efavirenzum, Imatinib*) have a narrow therapeutic range and a poorly predictable relationship between the dose and the concentration of the drug in the blood; the relationship can greatly vary among individuals.

Therapeutic drug monitoring (TDM) is a key concept in precision medicine [MW14]. The goal

of TDM is to avoid therapeutic failure or the toxic effects of a drug due to its insufficient or excessive concentration related to the between-patient variability. Recently, *TUCUXI* – an intelligent system for TDM – was developed [FCT+12]. By making use of embedded mathematical models, the software computes, based on a patient's parameters and previously observed concentrations, the maximum-likelihood individual predictions of drug concentrations from the population pharmacokinetic data. *TUCUXI* was developed to be used in medical practice, to assist clinicians in taking dosage-adjustment decisions in order to optimize drug concentration levels. However, the integration of software in the clinical workflow, and for research purposes, is a very challenging process.

The collection of population data from the daily use of the software is ideally suited to improve the existing models and to develop new models for drug candidates for TDM. However, patients' data are sensitive, studies have very different scopes, and data aggregation is time consuming.

We address the issues related to interoperability and data fusion for primary care and research in collaboration with CHUV, Lausanne University Hospital, Switzerland. In order to integrate *TUCUXI* in clinical practice, we studied the current clinical dataflow in CHUV. We also had at our disposal the data already collected in the framework of TDM for testing the interoperability between the different components of the healthcare infrastructure. This dataset was composed of two separate databases, one with 8898 records (called *Gentamicin_large*), and a second one with the extended schema, containing more health information within 224 records (called *Gentamicin_small*), in total, the database had 9122 medical entries. The data about pre-term and term newborns (treated with an antibiotic, *Gentamicin*) were collected in the neonatal intensive-care unit in CHUV; these data were used both for the treatment and for research purposes in the framework of the ISyPeM project[1]. These data were previously statically anonymized in the hospital so that it is impossible to re-identify patients (more details can be found in Chapter 6).

2.1.2 Requirements and Regulations in eHealth

The underlying concept of precision medicine is not new: healthcare can be individually adjusted based on the information about a person such as his genes, lifestyle and environment. For instance, for more than a century, transfusion patients have been matched with donors according to their blood types [Hod16]. Sharing genomic data, health records, and the experience of millions of people will improve precision medicine: the richer the databases are, the better patient care will become [B+16], due to advances in medical research.

Patients can benefit from data sharing not only in the future, but also in the short-term [CHL+12]. This is especially relevant in the case of chronic diseases or serious medical conditions, such as cancer or HIV, during the treatment and post-treatment monitoring, when the patient has to visit multiple hospitals and other primary-care institutions. A patient's

[1] http://www.nano-tera.ch/projects/368.php

medical history evolves over time; it has to be properly maintained and shared in order to help caregivers find the best treatment strategy for the patient.

An electronic health record (EHR) is a digital version of the traditional paper-based medical record for an individual. It includes demographics, medical history, medications, allergies, immunization status, lab test results, radiology images, vital signs, personal statistics such as age and weight, and billing information. EHRs are being adopted all over the world.

The Health Information Technology for Economic and Clinical Health (HITECH) Act of 2009 provided a series of incentives to encourage widespread EHR adoption. By mandating standards for storing and sharing EHRs, the HITECH Act aims to improve healthcare quality, safety, and efficiency. The European commission developed the eHealth action plan 2012-2020[2] that includes actions to increase digital health literacy of health professionals and patients, and puts a special focus on mobile health and on connecting data. According to the 2015 World Health Organization global survey on eHealth data, 27 out of 53 countries of the European Region have a national electronic health record system. Among them, 18 member states have legislations governing its use. Integrating the Healthcare Enterprise (IHE[3]) is an initiative by healthcare professionals and the industry to improve the way computer systems in healthcare share EHRs. IHE provides specifications, tools, and services that aim to achieve interoperability. IHE Integration Profiles, for instance, offers a common framework for vendors and IT departments to address clinical integration needs. IHE Integration Profiles promote the use of established healthcare standards such as HL7 and DICOM[4]. HL7 is a set of standards[5], formats and definitions for exchanging and developing electronic health records.

Although the HITECH Act, the EU action plan, and the IHE initiative have been effective in helping digitize medical records, patient privacy can create a barrier for the advancement of knowledge through data digitalization and sharing [SRLH05]. Indeed, as defined under US law, the information that falls into the Protected Health Information (PHI) category includes any information about health status, provision of health care, or payment for health care that is created or collected by a "Covered Entity" (e.g., medical service providers, health insurers) and that can be linked to a specific individual. In a nutshell, any part of a patient's medical record or payment history is considered to be PHI.

HIPAA (Health Insurance Portability and Accountability Act of 1996) is the US legislation that requires the establishment of a nationwide protection of patient confidentiality, security of electronic systems, and standards and requirements for electronic transmission of health information. The HIPAA Privacy Rule (effect April 14, 2003) establishes national standards to protect patient health information. The Privacy Rule defines how patient information is used and disclosed, gives the patients privacy rights and more control over their own health

[2] https://ec.europa.eu/digital-single-market/en/news/ehealth-action-plan-2012-2020-innovative-healthcare-21st-century

[3] https://www.ihe.net

[4] http://dicom.nema.org

[5] http://www.hl7.org

information, and outlines ways to safeguard PHI. The HIPAA Security Rule (effect April 21, 2005) controls the confidentiality of electronic protected health information (ePHI), the storage of ePHI, and the access to electronic information. In the European Union, the management of healthcare data is regulated by the Article 29 Working Party and the General Data Protection Regulation that from 2018 supersedes the EC Data Protection Directive 95/46/EC.

2.1.3 Required Privacy and Security Properties

In this section, we define the properties of a sensitive healthcare data-management system for the use cases previously defined.

For the system to comply with the legislation, each patient needs to provide consent to share his data both for primary care and research purposes. For primary care, the following security properties are essential: availability of the data, data integrity, and data confidentiality. These properties can be defined as follows:

- *Availability* refers to the ability to use the information or resource desired. Availability is an important aspect of reliability, as well as of system design [Bis03].
- *Integrity* refers to the trustworthiness of data or resources, and it is usually phrased in terms of preventing improper or unauthorized change. Integrity includes data integrity (the content of the information) and origin integrity (the source of the data, often called authentication) [Bis03].
- *Confidentiality* means preventing the disclosure of information to unauthorized individuals or systems [SFK07]. Although confidentiality refers to the data, privacy, as defined above, refers to the person and his right to decide to keep his personal data confidential.

Data anonymization can be an alternative to consent management when data are shared for research purposes. However, ambiguous definitions of terms and contradictory guidelines developed in the area of secondary use of medical data, even only within the European Union, raise problems from legal, ethical, and technical standpoints. Elger et al. [EII+10] discuss the main laws and guidelines that describe how to prepare data for use in medical research (including de-identification and pseudonymization). The authors propose the following definitions based on how the concepts of personal, identified, and identifiable data are formulated in the various legal documents:

- *Personal data* refers to data that are about an individual who can reasonably be identified or identifiable.
- *De-identification* is the process of removing (or modifying) identifiers from the personal data so that identification is not reasonably possible.

- *Pseudonymization* is the step where a pseudonym or code is added to this de-identified data.
- *Proportional* or *reasonable anonymity* applies to de-identified/pseudonymized data that cannot reasonably be used to identify specific individuals. (The concept of proportional or reasonable anonymity was first introduced in a document published by the World Health Organization in 2003).

In practice, traceability is required in addition to interoperability and compliance with legislation and policies that regulate management of the personal data and, in particular, protected health information. *Traceability of the data* can be defined as the ability to retain the identities of the origin of the data, the entities who accessed the data, and the operations performed on the data (e.g., updates) [DLLK+11]. The data traceability will be particularly useful in legal cases, as well as for the patient, in defining and enforcing his access-control policy, in allowing meaningful data aggregation for research purposes, and in enabling reproducibility of research.

2.2 Cryptographic Primitives

Cryptographic primitives are a set of mathematical techniques related to aspects of information security such as confidentiality, data integrity, entity authentication, and data origin authentication [MVOV96]. In this section, we present the primitives that are employed when designing and building systems for healthcare-data management and taking into account the requirements defined above.

Notations
If $\lambda \in N$, then 1^λ is the string a length of λ ones. The empty string is denoted ϵ. For a bitstring $a \in \{0,1\}^*$ we denote the bit-length of a by $||a||$. If $\mathscr{A}$ is a randomized algorithm, then $y \xleftarrow{\$} \mathscr{A}(\S)$ denotes the assignment to y of the output of $\mathscr{A}$ on input x when run with fresh random coins.

Unless noted, all algorithms are probabilistic polynomial-time (PPT), and we implicitly assume they take an extra parameter 1^λ as their input, where λ is the security parameter. We denote by $Y \xleftarrow{\$} \mathbf{F}(X)$ a *probabilistic* algorithm that takes an input X and outputs Y. A notation $Y \leftarrow \mathbf{F}(X)$ is used for a *deterministic* algorithm with input X and output Y.

The Decisional Diffie-Hellman Assumption:
Let GroupGen be a probabilistic polynomial-time algorithm that takes as input a security parameter (1^λ), and outputs a triplet (G,p,g) where G is a group of order p that is generated by $g \in G$, and p is an λ-bit prime number. Then, the Decisional Diffie-Hellman (DDH) Assumption states that the tuples (g, g^a, g^b, g^{ab}) and (g, g^a, g^b, g^c) are computationally indistinguishable, where $(G,p,g) \leftarrow \mathsf{GroupGen}(1^\lambda)$, and $a,b,c \in \mathbb{Z}_p$ are chosen independently and uniformly at random.

Hash functions

A *hash functions* takes a message as input and produces an output referred to as a hashcode, or simply hash. The function is deterministic and public, but the mapping should look "random". In other words, a hash function H maps bit strings of arbitrary finite length to strings of fixed length d: $H : \{0,1\}^\star \rightarrow \{0,1\}^d$. The function is many-to-one, implying that the existence of collisions (pairs of inputs with identical output) is unavoidable [MVOV96]. The Random Oracle model [CGH04] is an ideal model of the hash function. In this model, it is assumed that there exists an oracle H such that on input $x \in \{0,1\}^\star$, if H has not seen x before, then it outputs a new random value $H(x)$. Otherwise, it returns the previously output value. The random oracle gives a random value for all new inputs, and gives deterministic answers to all inputs it has seen before. A random oracle does not exist since it requires infinite storage, so in practice pseudo-random functions are used.

The basic idea of cryptographic hash functions is that a hash-value serves as a compact representative image (message digest) of an input string, and can be used as if it were uniquely identifiable with that string. A hash function (in the unrestricted sense) is a function h which has at least the following two properties:

(i) compression – H maps an input x of an arbitrary finite length, to an output $H(x)$ of a fixed length;

(ii) ease of computation — given H and an input x, $H(x)$ is easy to compute.

In addition, a ***secure* cryptographic hash function** has the following properties:

(1) One-way (pre-image resistance): Given $y \in \{0,1\}^d$, it is hard to find an x such that $H(x) = y$;

(2) Strong collision-resistance: It is hard to find any pair of inputs x, x' such that $H(x) = H(x')$;

(3) Weak collision-resistance (target collision resistance, 2nd pre-image resistance): Given x, it is hard to find x' such that $H(x) = H(x')$;

(4) Pseudo-random: The function behaves indistinguishably from a random oracle;

(5) Non-malleability: Given $H(x)$, it is hard to generate $H(f(x))$ for any function f.

There are many applications of hash functions including password storage verification of authenticity and integrity of data, and commitments in secure bidding protocols.

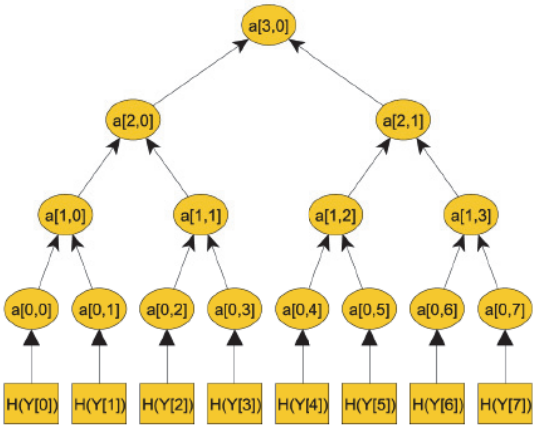

Figure 2.1 – Merkle tree with 8 leafs

Merkle Tree

Merkle Tree (also called hash tree) is a binary tree with nodes $a_{i,j}$, where i denotes the level of the node. The level of a node is defined by the distance from the node to a leaf: a leaf of the tree has level $i = 0$, and is represented by a hash of a message Y_k, and the root has level $i = n$. On Figure 2.1 we show an example of a tree with 8 leafs from [Bec08]. The nodes on each level are numbered from left to right, starting from zero, so that $a_{i,0}$ is the leftmost node of level i. In the Merkle Tree the hash values h_i are the leaves of a binary tree, so that $\mathsf{H}_k = a_{0,k}$. Each inner node of the tree is the hash value of the concatenation of its two children, so $a_{1,0} = \mathsf{H}(a_{0,0} \| a_{0,1})$ and $a_{2,0} = \mathsf{H}(a_{1,0} \| a_{1,1})$ [Mer80].

Such an organization of the hashes enables us to efficiently verify the integrity of data by storing only the root of the tree and the list of the input messages, Y_k. Any modification of the input data Y_k will result in a different hash stored at the leaf, and therefore, modification of the root of the tree. The main advantage of Merkle trees is that when one message is changed it is not necessary to compute a hash over all the data, as opposed to naive hashing. The number of required hash computations scales logarithmically in the number of data blocks. For instance, in the blockchain technology, the root of the Merkle tree is used for the efficient verification of the integrity of the transactions stored in a block [Oku14].

Symmetric encryption

Data confidentiality is one of the most basic security properties and is used in almost every security protocol. Confidentiality can be achieved through encryption: the standard approach for keeping sensitive data secret is to encrypt the data with a secret key that is only intended for the receivers to possess. Classical cryptosystems (also called single-key or symmetric cryptosystems) are cryptosystems that use the same key for encryption and decryption

Let $\{E_e : e \in K\}$ be a set of encryption transformations, and let $\{D_d : d \in K\}$ be the set of corresponding decryption transformations, where K is the key space. In *symmetric ciphers* encryption (e) and decryption (d) keys are essentially the same ($e = d = k, k \in K$), meaning

that encryption and decryption transformations are performed under the same secret key, k or the encryption key can be calculated from the decryption key and vice versa.

A *symmetric key encryption* scheme Enc^S is a set of algorithms ($\mathsf{Enc}^S.\mathsf{KeyGen}, \mathsf{Enc}^S.\mathsf{Encrypt}, \mathsf{Enc}^S.\mathsf{Decrypt}$) where:

$\mathsf{Enc}^S.\mathsf{KeyGen}(1^\lambda) \xrightarrow{\$} \mathsf{SK}^{\mathsf{symm}}$: Takes as input a security parameter (1^λ). It outputs a secure key $\mathsf{SK}^{\mathsf{symm}}$.

$\mathsf{Enc}^S.\mathsf{Encrypt}.(\mathsf{SK}^{\mathsf{symm}}, m) \xrightarrow{\$} \mathscr{C}$ Takes as input a secure key $\mathsf{SK}^{\mathsf{symm}}$, and a message m. It outputs a ciphertext $\mathscr{C}$.

$\mathsf{Enc}^S.\mathsf{Decrypt}.(\mathsf{SK}, \mathscr{C}) \rightarrow m/\perp$: Takes as input a secret key $\mathsf{SK}^{\mathsf{symm}}$ and a ciphertext $\mathscr{C}$. It outputs message m, or $\perp$, if ciphertext is invalid.

Public-key encryption

In 1976, Diffie and Hellman [DH76] proposed a new type of cryptography that distinguishes between encryption and decryption keys. One of the keys would be publicly known (public key, PK); the other (private key, SK) would be kept private by its owner. Classical cryptography requires the sender and recipient to share a common key, whereas public key cryptography does not. In order to send a secret message, a sender simply encrypts the message with the recipient's public key and sends it. The recipient can decrypt it using his private key. In *public-key encryption, PKE* a pair of associated encryption and decryption transformations (E_e, D_d) with the following property is considered: knowing E_e it is computationally infeasible, given a random ciphertext $c \in \mathscr{C}$, to find the message $m \in \mathscr{M}$ such that $E_e(m) = c$. This property implies that given e (public key, PK) it is infeasible to determine the corresponding decryption key d (private key, SK).

A public key encryption scheme Enc is a set of algorithms ($\mathsf{Enc.KeyGen}, \mathsf{Enc.Encrypt}, \mathsf{Enc.Decrypt}$) where:

$\mathsf{Enc.KeyGen}(1^\lambda) \xrightarrow{\$} (\mathsf{SK}, \mathsf{PK})$: Takes as input a security parameter (1^λ). It outputs a public key PK and a corresponding secret key SK.

$\mathsf{Enc.Encrypt}.(\mathsf{PK}, m) \xrightarrow{\$} c$: Takes as input a public key PK, and a message m. It outputs ciphertext c.

$\mathsf{Enc.Decrypt}.(\mathsf{SK}, c) \rightarrow m/\perp$: Takes as input a secret key SK and ciphertext c. It outputs message m or $\perp$ if ciphehertext is invalid.

In addition to confidentiality, public-key ciphers can provide data and origin authentication. If the sender enciphers the message using his private key, anyone can read it, but if anyone alters the ciphertext, the (altered) ciphertext cannot be deciphered correctly [Bis03].

Digital signature

Signature schemes is a fundamental cryptographic primitive for authentication, authorization, and nonrepudiation [Gol09]. A digital signature is a construct that authenticates both the origin and contents of a message in a manner that is provable to a disinterested third party. A party signs the message with her private key and other parties can check the signature with the public key.

A digital signature scheme Sig is a set of probabilistic polynomial-time algorithms ($\mathsf{Sig.KeyGen}$, $\mathsf{Sig.Sign}$, $\mathsf{Sig.Verify}$) where:

$\mathsf{Sig.KeyGen}(1^\lambda) \xrightarrow{\$} (\mathsf{SK}, \mathsf{VK})$: Takes as input a security parameter (1^λ). It outputs a verification key VK and a corresponding secret key SK. Message space $\mathscr{M}_{\mathbf{Sig}}$ is associated with VK.

$\mathsf{Sig.Sign}(\mathsf{SK}, m) \xrightarrow{\$} \sigma$: Takes as input a private key SK and a message m. It outputs a signature σ.

$\mathsf{Sig.Verify}(\mathsf{VK}, m, \sigma) \rightarrow 1/0$: Takes as input a public (verification) key VK, a message m and a signature σ. It outputs 1 for acceptance or 0 for rejection according to the input.

A digital signature scheme ($\mathsf{Sig.KeyGen}$, $\mathsf{Sig.Sign}$, $\mathsf{Sig.Verify}$) is called *correct*, if

$$\forall 1^\lambda, \forall (sk, pk) \leftarrow \mathsf{Sig.KeyGen}(1^\lambda) \forall m \in \mathscr{M}_{\mathbf{Sig}} : \mathsf{Sig.Verify}(m, \mathsf{Sig.Sign}(\mathsf{SK}, m), \mathsf{PK}) \xrightarrow{\$} 1.$$

Correctness and unforgettability of the signature scheme ensure integrity and authenticity of the data. Besides their use for signing digital documents and building classical cryptographic protocols such as key exchange, digital signature schemes are also important components of more advanced cryptographic protocols, such as blind signatures [Cha84], group signatures [BMW03], voting schemes [HS00], and anonymous credentials [BCKL08].

Functional encryption

Functional encryption is a paradigm in public-key cryptography that gives users more control over the amount of information that is revealed by a ciphertext to a given receiver [ABCP15]. It allows the receiver to decrypt only a result of the function (e.g., `sum`, or a weighted mean of a vector) applied to the encrypted data without having access to the input of the function (e.g., the terms of the resulting summation, or the coordinates of the vector). The functionality of this approach is based on the homomorphic properties of the ElGamal scheme [HX94] that allows to sum up encrypted values by multiplying corresponding ciphertexts.

We propose to use this scheme in order to preserve the $k-$*anonymity* property of the database in the construction of the evaluation phase (**EVAL**) of distributed one-step de-generalization protocol presented in Chapter 4. At this phase, the goal is to compute the sum of the records from different databases without learning the number of records stored in each database.

We use the definitions of the notion of the functionality F and functional encryption scheme FEnc over functionality F from [ABCP15]. A functionality F defined over (K, X) is a function F: $K \times X \rightarrow y \in \sum \cup \{\perp\}$ where K is the key space, X is the message space, $\sum$ is the output space, and $\perp$ is a special string not contained in $\sum$. The functionality is undefined for when the key or the message is not in the message space.

A functional encryption (FEnc) scheme FEnc for functionality F is a tuple $\mathsf{FEnc} = (\mathsf{FEnc.Setup}, \mathsf{FEnc.KeyDer}, \mathsf{FEnc.Encrypt}, \mathsf{FEnc.Decrypt})$ of 4 algorithms where:

$\mathsf{FEnc.Setup}(1^\lambda) \xrightarrow{\$} (\boldsymbol{mpk}, \boldsymbol{msk})$ outputs a public and a master secret keys for a security parameter λ.

$\mathsf{FEnc.KeyDer}(\boldsymbol{msk}, k) \xrightarrow{\$} \boldsymbol{skk}$: Takes as input a master secret key $\boldsymbol{msk}$ and a key $k \in K$. It outputs secret key SK_k.

$\mathsf{FEnc.Encrypt}(\boldsymbol{mpk}, x) \rightarrow \mathsf{SK}_k$: Takes as input public key $\boldsymbol{mpk}$ and a message $x \in X$. It outputs ciphertext c.

$\mathsf{FEnc.Decrypt}(\boldsymbol{mpk}, c, \mathsf{SK}_k) \rightarrow y \in \sum \cup \{\perp\}$ outputs resulting vector defined in the output space with respect to the functionality defined above.

Abdalla et al. in [ABCP15] present the functional encryption scheme for the inner-product functionality, meaning that decrypting an encrypted vector $\vec{x}$ with a key for a vector $\vec{y}$ will reveal only $\langle \vec{x}, \vec{y} \rangle$ and nothing else. The security of this scheme is based on the DDH assumption.

Formally, the functional encryption scheme for the *inner-product functionality* can be defined as follows: $\mathsf{IPEnc} = (\mathsf{Setup}, \mathsf{KeyDer}, \mathsf{Encrypt}, \mathsf{Decrypt})$ where:

$\mathsf{IPEnc.Setup}(1^\lambda, 1^l)$ samples $(G, p, g) \leftarrow \mathsf{GroupGen}(1^\lambda)$ and $\vec{s} = (s_1, \ldots, s_l) \leftarrow \mathbb{Z}_p^l$, and sets $\boldsymbol{mpk} = (h_i\ g^{s_i})_{i \in [l]}$ and $\boldsymbol{msk} = s$. The algorithm returns the pair $(\boldsymbol{mpk}, \boldsymbol{msk})$.

$\mathsf{IPEnc.Encrypt}(\boldsymbol{mpk}, x)$: Takes as input a master public key $\boldsymbol{mpk}$ and a message $\vec{x} = (x_1, \ldots, x_l) \in \mathbb{Z}_p^l$, chooses a random $r \leftarrow \mathbb{Z}_p$ and computes $c_0 = g^r$ and, for each $i \in [l]$, $c_i = h_i^r \cdot g_i^x$. The algorithm returns the ciphertext $\vec{c} = (c_0, (c_i)_{i \in [l]})$.

$\mathsf{IPEnc.KeyDer}(\boldsymbol{msk}, \vec{y})$: Takes as input a master secret key $\boldsymbol{msk}$ and a vector $\vec{y} = (y_1, \ldots, y_l) \in \mathbb{Z}_p^l$. It computes and outputs the secret key $sk_y = \langle y, s \rangle$.

$\mathsf{IPEnc.Decrypt}(\boldsymbol{mpk}, \vec{c}, sk_y)$: Takes as input a master public key $\boldsymbol{mpk}$, ciphertext $\vec{c} = (c_0, (c_i)_{i \in [l]})$ and secret key sk_y for vector $\vec{y}$. It outputs the discrete logarithm in basis g of $\prod_{i \in [l]} c_i^{y_i} / c_0^{sk_y}$:

$$\begin{aligned}\mathsf{IPEnc.Decrypt}(\boldsymbol{mpk}, \vec{c}, sk_y) &= \prod_{i\in[l]} \frac{c_i^{y_i}}{c_0^{sk_y}} = \frac{\prod_{i\in[l]} \left(g^{s_i r + x_i}\right)^{y_i}}{g^{r(\sum_{i\in[l]} y_i s_i)}} \\ &= g^{\sum_{i\in[l]} y_i s_i r + \sum_{i\in[l]} y_i x_i - r(\sum_{i\in[l]} y_i s_i)} \\ &= g^{\sum_{i\in[l]} y_i x_i} = g^{\langle \vec{x}, \vec{y} \rangle}\end{aligned}$$

The above scheme limits the expressiveness of the functionality that can be computed, because a discrete logarithm computation is required to recover the final inner-product value. In order to overcome this limitation and to generalize to other settings, the authors in [ABCP15] also present a generic scheme, whose security can be based on the semantic security of the underlying public-key encryption scheme under randomness reuse.

Let $\mathsf{E} = (\mathsf{E.Setup}, \mathsf{E.Encrypt}, \mathsf{E.Decrypt})$ be a PKE scheme with the following structural and homomorphic properties.

Structure. The secret keys in E are elements of a group $(\mathbb{G}, +, 0_{\mathbb{G}})$, public keys are elements of group $(\mathbb{H}, \cdot, 1_{\mathbb{H}})$, and the message space is $\mathbb{Z}_q$ for a prime q. In addition, we require the ciphertexts to consist of two parts ct_0 and ct_1. The first part ct_0 corresponds to a commitment $C(r)$ of the randomness r used for the encryption. The second part ct_1 is the encryption $E(pk, x; r)$ in a group $(\mathbb{I}, \cdot, 1_{\mathbb{I}})$ of the message x under the public key pk and the randomness r.

Linear Key Homomorphism. A PKE has linear key homomorphism (LKH, for short) if for any two secret keys $sk_1, sk_2 \in \mathbb{G}$ and any $y_1, y_2 \in \mathbb{Z}_q$, the component-wise $\mathbb{G}$ - linear combination formed by $y_1 sk_1 + y_2 sk_2$ can be computed efficiently only using public parameters, the secret keys sk_1 and sk_2 and the coefficients y_1 and y_2. This combination $y_1 sk_1 + y_2 sk_2$ also functions as a secret key to a public key that can be computed as $pk_1^{y_1} pk_2^{y_2}$, where pk_1 (resp. pk_2) is a public key corresponding to sk_1 (resp. sk_2).

Linear Ciphertext Homomorphism Under Shared Randomness. A PKE has linear ciphertext homomorphism under shared randomness (LCH, for short) if it holds that $E(pk_1 pk_2, x_1 + x_2; r) = E(pk_1, x_1; r) \cdot E(pk_2, x_2; r)$.

Then, functional encryption scheme for the inner-product functionality is defined as follows $\mathsf{IPEnc}^{\mathsf{E}} = (\mathsf{Setup}, \mathsf{KeyDer}, \mathsf{Encrypt}, \mathsf{Decrypt})$ where:

$\mathsf{IPEnc}^{\mathsf{E}}.\mathsf{Setup}(1^\lambda, l, B)$: Calls E's key generation algorithm to generate l independent (sk_1, pk_1), $\ldots, (sk_l, pk_l)$ pairs, sharing the same public parameters params. Then, the algorithm sets the functionality's key space K_l and message space X_l to $M = 0, \ldots, B - 1^l \subseteq \mathbb{Z}_q$ and returns $\boldsymbol{mpk} = (\mathsf{params}, pk_1, \ldots, pk_l)$ and $\boldsymbol{msk} = (sk_1, \ldots, sk_l)$.

$\mathsf{IPEnc}^{\mathsf{E}}.\mathsf{KeyGen}(\boldsymbol{msk}, \vec{y})$: Takes as input a master secret key $\boldsymbol{msk}$ and a vector $\vec{y} = (y_1, \ldots, y_l) \in M$. It computes sk_y as an $\mathbb{G}$ - linear combination of $(sk_1, \ldots, sk_n)$ with coefficients $(y_1, \ldots, y_l)$, namely $sk_y = \sum_{i\in[l]} y_i sk_i$.

$\mathsf{IPEnc}^{\mathsf{E}}.\mathsf{Encrypt}(\boldsymbol{mpk}, \vec{x})$: Takes as input a master public key $\boldsymbol{mpk}$ and a message $\vec{x} = (x_1, \ldots, x_l) \in$

M and chooses shared randomness r in the randomness space of E. It computes $ct_0 = \mathsf{E}.C(r)$ and $ct_i = \mathsf{E}.E(pk_i, x_i; r)$. Then the algorithm returns the ciphertext $\vec{c} = (c_0, (c_i)_{i\in[l]})$.

$\mathsf{IPEnc}^{\mathsf{E}}.\mathsf{Decrypt}(\boldsymbol{mpk}, \vec{c}, sk_y)$: Takes as input master public key $\boldsymbol{mpk}$, ciphertext $\vec{c} = (c_0, (c_i)_{i\in[l]})$ and secret key sk_y for vector $\vec{y}$. It returns the output of $E.\mathsf{Decrypt}(sk_y, (c_0, \prod_{i\in[l]} c_i^{y_i}))$. Correctness follows from the LCH property:

$$\begin{aligned}
\mathsf{IPEnc}^{\mathsf{E}}.\mathsf{Decrypt}(\boldsymbol{mpk}, \vec{c}, sk_y) &= E.\mathsf{Decrypt}(sk_y, (c_0, \prod_{i\in[l]} c_i^{y_i})) \\
&= E.\mathsf{Decrypt}(sk_y(c_0, \prod_{i\in[l]} \mathsf{E}.E(pk_i, x_i; r)^{y_i})) \\
&= E.\mathsf{Decrypt}(sk_y(c_0, \mathsf{E}.\mathsf{Encrypt}(\prod_{i\in[l]} pk_i^{y_i}, \sum_{i\in[l]} y_i x_i; r))) \\
&= \sum_{i\in[l]} y_i x_i.
\end{aligned}$$

Finally, the decryption is allowed because $(sk_y, \prod_{i\in[l]} pk_i^{y_i})$ is a valid key pair, due to the LKH property.

Secret-sharing scheme

A secret sharing scheme enables one to share a secret among n participants in such a way that only some sets of the participants, called allowed coalitions, can recover the secret, while any other sets of participants (non-allowed coalitions) cannot get any additional (i.e., a posteriori) information about the possible value of the secret. An (n, k)-threshold scheme is a particular case of a secret sharing scheme when any set of k or more participants can recover the secret exactly while any set of less than k participants gains no additional, that is, a posteriori, information about the secret.

Assume that the set S_0 of secrets is some finite field $GF(q)$ of q elements (hence q should be prime power) and that the number of participants of secret sharing scheme $n < q$. The dealer chooses n different nonzero elements (points) $x_1, \ldots, x_n \in GF(q)$ that are publicly known. To distribute a secret s_0, the dealer generates randomly coefficients $g_1, \ldots, g_{k-1} \in GF(q)$, forms the polynomial $g(x) = s_0 + g_1 x \cdots + g_{k-1} x^{k-1}$ of a degree less than k, and sends to the ith participant of the coalition the share $s_i = g(x_i)$. Clearly, any k participants can recover the whole polynomial $g(x)$ and, in particular, its zero coefficient (or $g(0)$), since any polynomial of degree l is uniquely determined by its values in $l+1$ points and Lagrange interpolation formula shows how to determine it. On the other hand, the point 0 can be considered as an evaluation point x_0, corresponding to the dealer, since $s_0 = g(0)$. Then the above consideration shows that for any given shares $s_1 = g(x_1), \ldots, s_{k-1} = g(x_{k-1})$ all possible values of s_0 are equally probable, hence the scheme is perfect.

Algorithm 1 demonstrates input, output, and the steps of the Shamir secret sharing algorithm. The reconstruction step is directly derived from polynomial interpolation and proceeds as follows: $s_0 = \sum_0^{k-1} s_i \cdot \beta_i$, where each $\beta_i = \prod_{j=0, j\neq i} \frac{-x_j}{x_i - x_j}$.

Algorithm 1 Shamir secret sharing

Input: A secret s_0, public $x_1, \ldots, x_n \in GF(q)$
Output: Shares $(x_i, s_i)_{i=0\ldots k-1}$
1: $(g_i)_{i=0\ldots k-1} \leftarrow Rand(1^\lambda)$
2: **for** $i = 0; i \leq d;\ i++$ **do**
3: $\quad s_i \leftarrow s_0 + g_1 x + \cdots + g_{k-1} x^{k-1}$
4: **return** $(x_i, s_i)_{i=0\ldots k-1}$

For some applications, it is convenient to have the maximal possible number n of participants equal to q, especially for $q = 2^m$. For Shamir scheme $n < q$ but the following simple modification allows to have $n = q$. Namely, the dealer generates a random polynomial of the form $f(x) = f_0 + f_1 x \cdots + f_{k-2} x^{k-2} + s_0 x^{k-1}$ and distribute shares $s_i = f(x_i)$, where the x_i are different but not necessary nonzero elements of $GF(q)$ [BK11].

We employ the Shamir secret-sharing algorithm with $n = q$ at the **INIT** phase of the distributed one-step de-generalization protocol in Section 4.4 for sharing a random number in the framework of the functional encryption scheme. Randomness is shared among the caregivers that contribute to the same equivalence class in the database that is being constructed for the research purposes.

Threshold encryption
A threshold public key encryption system is a public key system where the private key is distributed among n decryption servers so that at least k servers are needed for decryption. In a threshold encryption system, an entity, called the combiner, has a ciphertext $\mathscr{C}$ that it wishes to decrypt. The combiner sends $\mathscr{C}$ to the decryption servers, and receives partial decryption shares from at least k out of the n decryption servers. It then combines these k partial decryptions into a complete decryption of $\mathscr{C}$ [BBH06, SG98].

A Threshold Public Key Encryption (TPKE) system [BBH06] consists of five algorithms: TPKE = (TPKE.Setup, TPKE.Encrypt, TPKE.ShareDecrypt, TPKE.ShareVerify, TPKE.Combine).

TPKE.Setup(n, k, Λ): Takes as input the number of decryption servers n, a threshold k where $1 \leq k \leq n$, and a security parameter $\Lambda \in \mathbf{Z}$. It outputs a triple $(PK, VK, \boldsymbol{SK})$ where PK is the public key, VK is the verification key, and $\boldsymbol{SK} = (SK_1, \ldots, SK_n)$ is a vector of n private-key shares. Decryption server i is given the private-key share (i, SK_i) and it uses the private-key share to derive a decryption share for a given ciphertext. The verification key VK is used to check validity of responses from decryption servers.

TPKE.Encrypt(PK, m): Takes as input a public key PK and a message m. It outputs a ciphertext C.

TPKE.ShareDecrypt(PK, i, SK_i, C): Takes as input the public key PK, a ciphertext C, and one of the n private key shares in $\boldsymbol{SK}$. It outputs a decryption share $\mu = (i, \hat{\mu})$ of the encrypted message, or a special symbol $(i, \perp)$.

TPKE.ShareVerify($PK, VK, \mathscr{C}, \mu$): Takes as input PK, the verification key VK, a ciphertext $\mathscr{C}$, and a decryption share μ. It outputs valid or invalid. When the output is valid we say that μ is a valid decryption share of $\mathscr{C}$.

TPKE.Combine($PK, VK, \mathscr{C}, \{\mu_1, \ldots, \mu_k\}$): Takes as input PK, VK, a ciphertext $\mathscr{C}$, and k decryption shares $\{\mu_1, \ldots, \mu_k\}$. It outputs a cleartext m or $\perp$.

One of the applications of the threshold cryptosystems is implementation of a decryption policy. In our work we employ it to ensure the patient's privacy when performing the update of the research database.

2.3 Data Privacy Techniques

In this section we review existing data privacy techniques that are often employed for sharing, exchanging, and aggregating medical data: differential privacy (employed in case of statistical data release), k-*anonymity*, and pseudonymization.

Statistical Queries
Releasing of only *statistical data* or providing the possibility to perform only *aggregation queries* over the data (as it is proposed in [BHY15] and [HMRB15]) can guarantee the patients privacy. For instance, Bellika et al. presented an agent-based distributed system for privacy-preserving statistical query and processing of EHRs in [BHY15]. The role of the system in the proposed approach is to perform initialization and coordination of the distributed computation components among the sites participating in the computations. Ganta et al. [GKS08] proposed a solution against the privacy breach of overlapping population within multiple published datasets. The solution is based on the differential privacy model.

Differential privacy (DP) [DMNS06], is a privacy model that provides strong privacy guarantees independent of an adversary's background knowledge, computational power or subsequent behavior. This model requires that the outcome of any analysis should not overly depend on a single data record. It follows that even if a user had opted in the database, there would not be a significant change in any computation based on the database. Therefore, this assures every record owner that any privacy breach will not be a result of participating in a database. However, utility of the anonymized results provided by DP algorithms could be quite limited, due to the amount of noise that have to be added to the output, or because the data utility can only be guaranteed for a restricted type of queries [SCDFSM14]. Therefore, in the research studies where having just a result of a query is not sufficient, differential privacy can not be used.

It has been shown in [FES+17] that DP can be successfully employed in the distributed settings of the multiple databases DB_1,... DB_n when performing the SQL queries of the form `SELECT SUM(*)/COUNT(*)FROM` DB_1,... DB_l `WHERE * AND/OR * GROUP BY *`, where $l \leq n$ and $*$ denotes an arbitrary number of attributes.

Fredrikson et al. in [FLJ+14] evaluated the effectiveness of DP for building private versions of pharmacogenetic models. The authors conclude that current DP mechanisms do not simultaneously improve genomic privacy while keeping desirable clinical efficacy, thus suggesting the need for new mechanisms to be developed. Taking into account the use case scenario of therapeutic drug monitoring defined in Section 2.1.1 we have chosen to employ $k-$*anonymity* model for dynamic utility-preserving data aggregation in the distributed settings.

k* – *anonymity

$k-$*anonymity* model is widely used to guarantee a certain level of privacy against re - identification attacks (identity disclosure), by ensuring that any record in an anonymized dataset is indistinguishable with respect to a set of predefined attributes, so-called quasi-identifiers, from at least $k-1$ records in the dataset. For different kinds of attributes (single-valued/set-valued; numerical/categorical) a variety of algorithms have been proposed [Sam01, FWY05, BA05, KPE+12, LDR05, LDR06]. Poulis et al. [PLGDS13] propose $(k, k^m)-$*anonymity* model that ensures anonymity in case of combination of both relational (single-valued) attributes and transaction attributes (set-valued) with bounded information loss in one attribute type and minimal information loss in the other. This approach could be particularly applied in healthcare data management, as medical datasets often contain both relational (e.g., age, gender), and transaction (e.g., diagnosis codes) attributes.

Several models, built on top of $k-$*anonymity*, were proposed to address the threats that occur when an attacker can infer with high probability that an information about an individual is present in the published data (the membership attacks), and when an individual is associated with information about their sensitive attributes (sensitive information disclosure) [GDLS14]. These models include l-diversity [MGKV06], t-closeness [LLV07], δ-presence [NAC07]. A recent review [GDLS14] describes existing models that are used in against re-identification and membership attacks and against sensitive information disclosure in healthcare.

Pseudonymization

Pseudonymization is a process when a pseudonym or code is added to the de-identified data. In medical settings, this technique could be employed when storing encrypted medical data in the cloud for the primary care, or for medical research. Pseudonymization can be used to create unambiguous pseudonym for the patient [Lo 07], or multiple pseudonyms as in [XC14].

There exists a line of work on pseudonymization in healthcare, aiming at de-sensitizing patient records [NLL07, AKRKG13, DMDMRF08, PRDS05, NH11, XC14, Lo 07]. The main focus in these works is to derive pseudonyms from unique patient identiers, such that the pseudonyms do not reveal any information about the patient anymore, yet allow de-anonymization by a trusted party (or a combination of several semi-trusted parties). In these solutions, pseudonym generation must be repetitively unambiguous to preserve the correlation between all pseudonymized records and allow their linkability.

While it is clearly beneficial to be able to link pseudonymized data originated from different sources and corresponding to the same person (for instance, for the purposes of med-

ical research), the context information and metadata can be leveraged to fully re-identify pseudonymized datasets, as it was demonstrated for credit card transactions (considered to be anonymized) [DMRS+15] and in the Netflix incident [NS08]. To ensure the users's privacy, for instance in reputation or e-voting systems, it can be undesirable to link the pseudonyms. AnonRep [ZWC+16] is the first practical anonymous reputation system maintaining the unlinkability and anonymity of users' historical activities. AnonRep uses verifiable shuffles and linkable ring signatures, with a multi-provider deployment architecture.

To provide controlled linkability, Camenisch and Lehmann proposed a combined approach: pseudonyms should be unlinkable by default, yet preserve the correlation which allows to re-establish the linkage only if necessary and based on the policy specified via a (potentially untrusted) converter [CL17]. The converter establishes individual pseudonyms for each server derived from a unique main identifier that every user has, but without learning the derived pseudonyms. The converter is still the only authority that can link different pseudonyms together, but it does not learn the particular user or pseudonym for which such a translation is requested. To construct such framework, the authors use dual-mode signatures (which allow one to sign messages in the plain as well as when they are contained in an encryption), (verifiable) pseudorandom functions, and homomorphic encryption.

2.4 Distributed consensus and blockchain technology

Blockchain is a peer-to-peer distributed ledger technology that provides a shared, immutable, and transparent append-only register of all the transactions that happen in the network. It is secured using cryptographic primitives such as hash function, digital signature, and encryption [Nak08]. The data in the form of transactions, digitally signed and broadcasted by the participants, are grouped into blocks in the chronological order and time-stamped. A hash function is applied to the content of the block and forms a unique block identifier, which is stored in the subsequent block. Due to the properties of the hash function (result is deterministic and can not be reversed), by hashing the block content again and comparing it with the identifier from the subsequent block, one can easily verify if the content of the block was modified. The blockchain is replicated and maintained by every participant. With this decentralized approach there is no need for setting up a single trusted centralized entity for managing the registry. The participants will notice a malicious attempt to tamper the information stored in the registry, hence the immutability of the ledger is guaranteed.

Adding a new block to the existing ledger is defined by the **consensus** protocol employed in the implementation of the blockchain technology. Based on how the identity of a participant and its right to participate in the consensus are defined within a network, one could distinguish between *permissionless* and *permissioned* blockchain systems, and *public* and *private* among the latter. A permissionless system is one in which the participants' identities are either pseudonymous or even anonymous [Swa15], every user may participate in the consensus protocol, and, therefore, append a new block to the ledger. In contrast, in case of a permissioned

blockchain, identities of the users and the rights to participate in the consensus (right to write to the ledger and/or validate the transactions) are controlled by a membership service. A permissioned blockchain is *public* when anyone can read the ledger, but only pre-defined set of users can participate in the consensus, and *private*, when even the right to read the ledger is controlled by the membership/identity service.

Many blockchain systems can execute arbitrary tasks, typically called **smart contracts**[6], written in a domain-specific or a general-purpose programming language. Below we are going to review different types of consensus protocols and blockchain systems that employ these protocols. Then, we introduce two of the most well-known mature *implementations of the blockchain technology* that are massively employed in order to build applications on top of blockchain: **Ethereum** [But14] and **Hyperledger**[7].

2.4.1 Distributed consensus protocols in blockchaian systems

In this section we review distributed consensus protocols and membership mechanisms used in blockchain systems.

Consensus protocols in blockchain
As summarized by Schneider in [Sch90], the task of reaching and maintaining consensus among distributed nodes can be described with two elements: (i) a (deterministic) state machine that implements the logic of the service to be replicated; and (ii) a consensus protocol to disseminate requests among the nodes, such that each node executes the same sequence of requests on its instance of the service. In the literature (e.g., [Sch90, CV17]), consensus means traditionally only the task of reaching an agreement on one single request, whereas atomic broadcast [HT93] provides an agreement on a sequence of the requests, as needed for the state-machine replication. Since there is a close connection between the two (a sequence of consensus instances provides atomic broadcast), in the context of blockchains, the term consensus often stands for atomic broadcast.

Consensus in permissioned blockchain
The form of consensus relevant for permissioned blockchain is technically known as *atomic broadcast* [CV17]. Atomic broadcast ensures the following properties: validity, agreement, integrity, and total order. In other words, it ensures that each correct node outputs or delivers the same sequence of messages through the deliver events. The most important and most prominent way to implement atomic broadcast in distributed systems prone to $t \leq n/2$ node crashes is the family of protocols known as *Paxos* [Lam98] and *Viewstamped Replication (VSR)* [OL88]. These protocols were discovered independently, but their core mechanisms are built on the same ideas. These protocols progress in a sequence of views or epochs, with a unique leader for each view that is responsible for progress. If the leader fails or, if the other

[6] Smart contract and chaincode logic concepts are quite close, therefore, we use the former when talking about programmable contract, or a set of rules, when discussing blockchain technology in general.
[7] https://github.com/hyperledger/

nodes suspect that the leader has failed, they can replace the current leader by moving to the next view with a fresh leader. Raft [OO14] is a specialized variant of a distributed consensus protocol from the same family; it was developed with the aim of simplifying the understanding and the implementation of Paxos.

More recently, consensus protocols for tolerating *Byzantine* nodes have been developed, where nodes may be subverted by an adversary and act maliciously against the common goal of reaching consensus."The Byzantine generals problem" can be formulated as finding an algorithm that ensures that the honest participants of the network of nodes will reach an agreement by exchanging messages even if some nodes fail, collude, or send corrupted messages. It is proven that no solution to the Byzantine generals problem can tolerate more than $f < n/3$ byzantine nodes in the network [LSP82]. *PBFT* is a partially synchronous protocol for Byzantine state-machine replication that was presented in [CL02]. It can be seen as an extension of the previously proposed Paxos/VSR family of consensus protocols, and also uses a progression of views and a unique leader within every view. A more detailed analysis of the consensus protocols used in some prominent permissioned blockchain platforms could be found in a recent review [CV17].

Ripple consensus protocol was proposed in [SYB14] as a low-latency variant of PBFT. In this protocol, it is assumed that each node would declare on its own unique node list (UNL) – a list of nodes it trusts – instead of accepting a global assumption on which node collusions the protocol tolerates. The process of advancing the distributed ledger is controlled by so-called validating nodes. Every few seconds the validating nodes start to create a new entry, and iteratively vote in rounds on its content. It is required that $4/5 \times n$ of all n validator nodes must be correct for maintaining correctness. This corresponds to tolerating $f \leq n/5$ malicious nodes.

Stellar is blockchain framework that is also based on validator nodes and uses a protocol called *federated Byzantine agreement* [Maz15]. Each validator declares its own convincing-set (that is called "quorum slice" and is similar to Ripple's UNL) that must overlap with the convincing-sets of other nodes to prevent forks. A node accepts a "vote" or a transaction for the ledger when a threshold of nodes in its convincing-set confirm it. This approach introduces some amount of centralization, similar to the Ripple's limited recommended set of validating nodes.

Consensus in permissionless blockchain

Nakamoto consensus [Nak08] realizes a replicated state machine abstraction, where nodes in a permissionless network reach agreement about a set of committed transactions as well as their ordering. The protocol relies on chaining blocks of transactions. Nodes express their acceptance of the block by working on creating the next block in the chain, using the hash of the accepted block as the previous hash. Nodes always consider the longest chain to be the correct one and will keep working on extending it. If two nodes broadcast different versions of the next block simultaneously, some nodes may receive one or the other first. In this case, they work on the first one they receive, but save the other branch in case it becomes longer;

the nodes that were working on the other branch will then switch to the longer one. New transaction broadcasts do not necessarily need to reach all nodes. As long as they reach many nodes, they will get into a block before long. Block broadcasts are also tolerant of dropped messages. If a node does not receive a block, it will request it when it receives the next block and realizes it missed one.

Membership mechanisms for non-Byzantine consensus
Since in permissionless settings the nodes' identities are not known a priori, it is imperative to defend against a Sybil attack where an attacker makes up arbitrarily many identities to outvote honest nodes. Nakamoto consensus critically relies on proofs-of-work to roughly enforce the idea of "one vote per hashpower".

The *Proof-of-work (PoW)* consensus protocol was presented in [Nak08] in the first application of blockchain technology: the Bitcoin cryptocurrency. PoW is based on "mining": a process of searching for a nonce – a random number that is stored in every block – so that the resulting hash of a new valid block satisfies certain requirements. These requirements set the difficulty threshold for the process of finding the nonce and determine the average number of hashes needed to mine one block. The difficulty threshold impacts the amount of energy to be spent to find such nonce. In 2013 the amount of energy used by Bitcoin mining was already comparable to the Irish national energy consumption [OM13]. Existing PoW blockchains can achieve throughput of not more than 60 transactions per second without significantly affecting the blockchain's security [GKW+16]. These two findings show that PoW can negatively impact the system scalability and overall throughput [Vuk15].

It is also possible for two valid blocks to be found at approximately the same time (depending on network latency). This leads to a temporary *fork* during which there are two equal-length chains. Miners can choose either fork in this scenario. Due to the random nature of the computational puzzle, one blockchain will eventually be extended further than the other, at which point all miners should adopt it.

Many researchers have challenged various aspects of the Bitcoin system, while trying to address these issues. The following modifications in PoW core operation have beed proposed: the modification of the block generation rate and of the basic primitives (e.g., a number of alternative proof-of-work implementations have been proposed using functions like scrypt [Per09], lyra2 [SJAA+14]). One of the most notable among more radical modifications is the GHOST protocol, which was suggested in [SZ15]. This protocol is based on the principle that all mined blocks should matter in the chain selection process, even those that did not end up in the main chain. In order to achieve this, players store a tree of all mined blocks they have received, and then using the greedy heaviest observed subtree (*GHOST*) rule, they pick which chain to mine. In [KP16], the authors present the first formal security proof of the GHOST blockchain protocol, in particular, they prove that GHOST is a robust transaction ledger that satisfies liveness and persistence. Persistence guarantees that if an honest party reports a transaction x blocks deep, then this transaction will be always reported, in the same position

and the depth equal to x or more then x, by all honest nodes. Liveness guarantees that if all honest parties attempt to insert the transaction in the ledger, then after u rounds, an honest party will report it x blocks deep in the ledger.

Proof-of-Stake (PoS[8] and Proof-of-Burn (PoB[9])), or *virtual mining mechanisms*, have been recently proposed as alternatives to PoW. Instead of having participants mine by exchanging their wealth for computational resources (which are then exchanged for mining rewards), in virtual mining, participants could exchange their wealth directly for the ability to append a new block to the ledger [BMC+15]. For example, in PoS, the selection of a participant that will create a new block is based on the amount of tokens owned by the participant. Ouroboros [KRDO17] is the first protocol based on PoS with the rigorous security guarantees. However, the authors in [KRDO17] assume that the network is highly synchronous. Moreover, a significant number of epochs most of the randomly selected stakeholders are incorruptible. It is assumed that these stakeholders form a committee that is then responsible for executing the coin-flipping protocol to produce the randomness for the leader-election process.

Proof-of-Burn (PoB) is based on the amount of tokens provably destroyed (sent to an unspendable address). However, providing a rigorous argument for or against the stability of virtual mining remains an open problem [BMC+15].

Hybrid consensus protocols

Membership mechanisms for non-Byzantine consensus could be combined with classical permissioned consensus protocols to achieve a responsive permissionless consensus protocol. *Hybrid consensus* combines classical-style and Nakamoto-style consensus protocols. Hybrid consensus relies on the Nakamoto consensus to reelect committees over time, where each committee consists of recently online miners. Then, to confirm the transactions, each committee will execute a classical, partially synchronous consensus instance (e.g., PBFT)[PS17].

Bitcoin-NG [EGSVR16] is another Bitcoin-like system that separates blocks into two categories, namely keyblocks and microblocks, reflecting the fact that transaction serialization and leader election may be separated [SZ15]. Keyblocks are generated using PoW-based GHOST mechanism and are used to securely select the leaders that then will generate and sign the microblocks that contain the transactions. This leads to an increase of the transactions throughput. However, for transactions to be confirmed it is still required to wait for certain amount of keyblocks to be added to the blockchain, and temporary forks can not be avoided. Moreover, in case of a misbehavior of a current leader, invalid transactions could be added to the blockchain.

Building on ideas proposed in [EGSVR16] and decoupling transaction verification from leader election to achieve a higher transaction throughput, *ByzCoin* [BKKJ+17] uses a group signature scheme called CoSi [STV+16] to reduce per-round communication complexity and signature verification of PBFT. CoSi combines Schnorr multi-signatures with communication trees to

[8] https://bitcointalk.org/index.php?topic=%2027787.0%2012.
[9] http://en.bitcoin.it/wiki/Proof_of_Burn

allow scalability to thousands participants. It involves four rounds of communication, at the end of which a collective signature is generated.

Algorand [GHM+17] provides a free of forks distributed ledger following a Byzantine agreement-per-block approach that can withstand adaptive corruptions. Block proposers and voting committee members are elected not based on the POW-based algorithm but on member's stake through Verifiable Random Function [MRV99]. The weight is assigned to the users based on the amount of cryptocurrency they possess to avoid Sybil attacks. It is then required that over 2/3 of cryptocurrency is owned by honest users to avoid forks and double-spending.

2.4.2 Blockchain Technology Implementations with Smart Contract Functionality

Blockchains may execute arbitrary, programmable transaction logic in the form of smart contracts, as exemplified by Ethereum[10]. The scripts in Bitcoin were a predecessor of the concept. A smart contract functions as a trusted distributed application and gains its security from the blockchain and the underlying consensus among the peers. This resembles the well-known approach of building resilient applications with state-machine replication (SMR) [Sch90]. However, blockchains differ from traditional SMR with Byzantine faults in the following: many distributed applications run concurrently; applications may be deployed dynamically and by anyone; and the application code is untrusted, potentially even malicious [ABB+18].

Below we describe in more detail the implementations of the blockchain technology (i.e., **Ethereum** and **Hyperledger** that provide a chaincode functionality, and, therefore, are massively employed in order to build applications on top of blockchain.

Ethereum

Ethereum [But14] is an implementation of a permissionless programmable blockchain that enables any user to create and execute the code of arbitrary algorithmic complexity on the Ethereum platform: Ethereum Virtual Machine (EVM). EVM can be seen as a large decentralized computer. "Accounts" of two types could be created on EVM. Externally owned account (EOA) is an account controlled only by a private key of a user. The owner of the private key associated with the EOA can remain anonymous (up to a certain degree) and has the ability to send messages. Contract account is the second type of accounts that can be seen as an autonomous agent that lives in the Ethereum execution environment and is controlled by its contract code: smart contract. Smart contract is used to encode arbitrary state transition functions, allowing users to create systems with different functionalities by transforming the logic of the system into the code. In case of public blockchain (such as Ethereum), smart contracts and all the transactions are public.

In Ethereum, transaction processing is Turing-complete and it can be used to implement any public functionality in a distributed way, but the code execution must be paid. The transaction price limits the number of computational steps for the code execution in order to prevent

[10] http://ethereum.org/

infinite loops or other computational wastage. Users can participate in the consensus process to obtain the tokens in order to pay for the transaction execution. In Ethereum, the consensus is achieved by using GHOST – modified proof-of-work (PoW) mechanism[11].

In order to avoid issues of network abuse, all programmable computations in Ethereum are subject to fees. The fee schedule is specified in units of gas. Thus any given fragment of programmable computation (i.e., creating contracts, making message calls, utilizing and accessing account storage, and executing operations on the virtual machine) has an universally agreed cost in terms of gas [Woo14].

Ethereum provides excellent scalability (in terms of number of nodes and clients), but has a limited transaction throughput. In 2016, typical Ethereum throughput was fewer than 20,000 transactions per day, i.e., about 0.2 tx/s on average [Vuk15].

Hyperledger Fabric

Hyperledger Fabric – an implementation of a permissioned blockchain – is an open source blockchain initiative hosted by the Linux Foundation. Recently available current version of Hyperledger Fabric (v1.0) brought some updates and improvements to the first stable release of Hyperledger Fabric (v0.6). We first describe the architecture of the Fabric v0.6, as this is the implementation employed in the solution presented further in this thesis. Then, we provide an overview of the specifics of Fabric v1.0.

In Hyperledger Fabric v0.6 there are two kinds of peers: validating and non-validating ones. A validating peer is a node on the network responsible for running consensus, validating transactions, and maintaining the ledger. A non-validating peer is a node that can be seen as a proxy to connect clients (issuing transactions) to validating peers. A non-validating peer does not execute transactions but it can verify them.

Hyperledger Fabric v0.6 has a modular architecture that allows plugging in different consensus mechanisms, including PBFT [CL02]. Transactions are validated through the execution of the chaincode and given that strictly not more then 1/3 of nodes behave arbitrarily and all others execute the chaincode correctly. In order to prevent divergence of the state of the peers, when employing PBFT consensus, it is important to ensure that chaincode transactions are deterministic [Cac16]. Hyperledger Fabric v0.6 relies on PBFT, hence the scalability in terms of nodes is limited (tested only up to 20 nodes) [Vuk15].

Hyperledger Fabric v0.6 contains a security infrastructure for authentication and authorization. It supports enrollment and transaction authorization through public-key certificates, and confidentiality for chaincode is realized through in-band encryption. In Hyperledger, smart contracts are implemented by the chaincode that consist of Logic and associated World state (State). Logic of the chaincode is a set of rules that define how the transactions will be executed and how the State will change. The State is a database that stores the information in a form of key-value pairs, where the value is an arbitrary byte array. The State also contains the

[11] http://www.ethdocs.org

block number to which it corresponds. The ledger manages the blockchain by including an efficiently cryptographic hash of the State when appending a block. This enables efficient synchronization if a node was temporary off-line, minimizing the amount of stored data at the node.

Membership services manage identity, privacy, and confidentiality in the network. A user is assigned a username and a password that will be used to issue the enrollment certificate (ECert) to identify every registered user. It is possible to use different transaction certificates (TCert) associated with the same ECert for every transaction to ensure the unlinkability of the transactions (a mapping between TCert and Ecert are only known to the membership service).

In Hyperledger Fabric v1.0 the architecture is extended with an additional special role of endorsing peer to separate the functions of validation among endorsers and consensus nodes. Confidential state on blockchain (seen only by endorsers) could be defined. Ordering-service node, or orderer, introduced in v1.0, is a node running the communication service that implements a delivery guarantee, such as atomic or total order broadcast. The ordering service can be implemented in different ways: ranging from a centralized service to distributed protocols that target different network and node fault models.

The proposed architecture of Hyperledger Fabric v1.0 aims at addressing the following challenges of permissioned blockchains, such as improving the scalability in the number of participant nodes, eliminating non-deterministic transactions, enabling multiple providers of Membership Services (to distribute the centralized management of identity, privacy, and confidentiality).

2.5 Agent Coordination in Distributed Environment

In this section we describe principles of multi-agent systems and some of the existing communication paradigms employed for agents coordination.

2.5.1 Principles of Multi-Agent Systems

An agent can be rationalized as an autonomous entity observing the surrounding environment through a perception layer, and possibly interacting with it, as well as with other agents. Self-developed or induced objectives (both pre-programmed decisions and dynamic ones) drive the agent choices while trying to maximize its goals and performance [RN03]. These intelligent agents are also able to extend/update their knowledge base, thus renewing their plans to achieve the desired goals [RN03].

Multi-agent systems (MAS) are generally composed of loosely coupled agents interconnected and organized in a network, each of them having the ability to solve problems and attain its goals by interacting with each other through collaboration, negotiation, and competition patterns [RN03]. While this general understanding of MAS may imply different degrees of

cooperation among agents, depending on the type of application and agent interaction model, very different behaviors can be observed in multi-agent systems. These may include concepts related to knowledge (data) sharing among agents, message-passing strategies, agreement and consensus, reputation and trust among agents, voting systems, agent identity management, and many more.

The natural abstraction of MAS, in biological and societal terms, supports the robustness of their mechanisms and behaviors. For example, Zambonelli and Omicini in [OZ99] assert the affinity of *ant foraging* and the *agents' mobility* in finding information within a distributed P2P network, and the similarity of social phenomena like the information propagation in social networks and routing algorithms. Social and natural phenomena with negotiation-based interactions and social conventions [COZ99, MT95] have been exploited extensively, shaping the multi-agent system paradigm.

The proactiveness and the possibility of performing dynamically intelligent behaviors with a high degree of autonomy are some of the most important features of MAS. Furthermore, MAS are particularly appreciated in the case of failure handling or resource optimization where required.

Finally, regardless distribution and dimensions, although broadly appreciated, MAS autonomy and flexibility still generate minor concerns about possible evolution in undesired behaviors of inferences and plans. Moreover, depending on the cooperative/competitive nature of the community factors such as *trust* [RHJ04] and *reliability* [CMS+17] are still open challenges heavily affecting the MAS pillars: *(i)* agent local scheduler, *(ii)* communication protocol, and *(iii)* negotiation protocol.

2.5.2 Agent coordination

There exist a number of communication paradigms that are being employed for coordination in multi-agent systems: publish/subscribe scheme, message passing, remote invocations, notifications, shared spaces, and message queuing to name a few. Below we describe the communication models that have been employed in this thesis, namely, publish/subscribe paradigm and TuCSoN (Tuple Centres Spread over the Network) coordination model.

Publish/Subscribe paradigm

The *Publish/Subscribe paradigm* allows one to deliver the data from the producers of the data (publishers) to consumers of the data (subscribers) in a distributed environment in a decoupled fashion [GSAA04]. In other terms, producers publish information on a software bus (an event manager) and consumers subscribe to the information they want to receive from that bus. This information is typically denoted by the term *event* and the act of delivering it by the term *notification*.

The basic system model for publish/subscribe interaction relies on an event notification service (publish/subscribe broker) providing storage and management for subscriptions and

efficient delivery of events. Such an event service represents a neutral mediator between publishers, acting as producers of events, and subscribers, acting as consumers of events. Subscribers register their interest in events by typically calling a `subscribe()` operation on the event service, without knowing the effective sources of these events. This subscription information remains stored in the event service and is not forwarded to publishers. The symmetric operation `unsubscribe()` terminates a subscription.

To generate an event, a publisher calls a `publish()` operation. The event service propagates the event to all relevant subscribers. Every subscriber will be notified of every event conforming to its interest (obviously, failures might prevent subscribers from receiving some events). The decoupling that the event service provides between publishers and subscribers can be decomposed along the following three dimensions: space decoupling, time decoupling,and synchronization decoupling [EFGK03].

TuCSoN coordination model

TuCSoN (Tuple Centres Spread over the Network) is a model for the coordination of distributed processes, as well as of autonomous, intelligent and mobile agents [OZ99]. Interaction between agents within TuCSoN coordination model is happening through shared tuple spaces that can be seen as a shared system such as blackboard system [Gel85]. Using the tuple center, an agent can insert (`out` operation), read (`rd` operation) and consume (`in` operation) tuples. The templates of the tuples need to be specified with respect to their structure, or the ontology model needs to be employed to interpret the information transferred by the tuples. In order to establish interaction between agents, coordination rules can be set up. ReSpecT [DNO98] – a first-order logic language – allows one to define the behavior of the tuple centers. The reaction rules syntax is defined as follows: `reaction (action, conditions, react)`, where `action` is an operation that was performed at the tuple center, `conditions` are the conditions that need to be verified before the execution of `react`, that describes the events caused by the `action` if the `conditions` were satisfied.

2.6 Related Work

In this section, we first discuss recent works that apply the blockchain technology for healthcare data-management. Second, we discuss the algorithms and platforms employed in existing frameworks for medical data management. Finally, we analyze existing pharmacokinetic tools for automatization of the process of therapeutic drug monitoring, from the perspectives of their functionality and possibility to provide integration in clinical data-flow and data aggregation for the research.

2.6.1 Traceability of Medical Data Using Blockchain

The possibility of using blockchain for healthcare data management has recently raised a lot of attention both in industry and academia [KKOM17, AEVL16, YWJ+16, BI16]. Yue et al. claim to be the first to import blockchain into the design of a healthcare system [YWJ+16]. They presented the architecture of a Healthcare Data Gateway application for easy and secure control and sharing of medical data between different entities that may use the patient's data. However, the system has not been implemented yet. The possibility to share data for research purposes is only sketched in the paper without security evaluation. Jenkins et al. proposed to use blockchain technology for a multi-factor authentication in a specific research scenario (medical data-analysis with functional biomarkers) that involves biometric and biomedical data [JKT+15]. A recent review [KKOM17] provides an extensive list of studies and ongoing projects that focus on exchanging patient care data using blockchains to improve medical record management, to conduct clinical studies, and to support healthcare financing tasks. The authors describe key benefits of using blockchain technology in health and discuss potential problems and challenges to be considered when adopting permissionless blockchain technology (e.g., speed and scalability, confidentiality, the treat of a 51% attack).

To the best of our knowledge, MedRec [AEVL16] is the first and the only functioning prototype that has been proposed until now. The authors presented a system based on Ethereum smart contracts for an intelligent representations of existing medical records that are stored within individual nodes on the network. As MedRec is based on permissionless blockchain implementation, it faces the challenges listed by Kuo et al. [KKOM17] and mentioned above, in addition to the management of the transaction fees and "mining".

2.6.2 Exchange and Aggregation of Medical Data

Comparative effectiveness research, (CER[12]) is the conduct and synthesis of systematic research comparing different interventions and strategies to prevent, diagnose, treat, and monitor health conditions. In an effort to address a demand for an inter-institutional CER there have been new designs and implementations of informatics platforms that provide access to electronic clinical data. Sittig et al. [SHB+12] provide an overview of six platforms pro-

[12] https://www.nlm.nih.gov/hsrinfo/cer.html

posed as a result of collaborative work among different organizations such as hospital systems, pharmacies, healthcare players, and laboratory organizations.

Only one platform among six that are listed in [SHB+12] provides publicly available data. However, this data can only be used for healthcare quality assessment. Another platform described by Sittig et al. is presented in the survey at its planning stage and we could not find any additional information available. Four other solutions provide platforms for research projects to be conducted in collaboration between selected medical centers on a study-by-study basis without support of dynamicity. In this case, an access to the data is granted only to the group of people involved in the particular project, with an exception for the project i2b2 [MWM+10], where a de-identified training data set can be accessed from the local network of the organization that hosts the platform.

Elger et al. provide an overview of technical, practical, legal, and ethical aspects of secondary data-use and discuss their implementation in the multi-institutional @neurIST project [EII+10]. In the framework of this project, the authors propose a strategy of federating data sources in the clinical institutions for the research based on a real-life example. The authors also list security vulnerabilities, including the possibility of cracking the proposed pseudonym generation mechanism, dependence on a trusted third party, and the possibility of establishing an indirect identification. However, they do not provide any solutions to these problems. Moreover, this approach only allows using data in the framework of a particular research project.

SciPort is a web-based collaborative biomedical data sharing platform that has been proposed by Wang et al. [WVNL14] to support data-sharing across distributed organizations. SciPort uses a central server-based data-sharing architecture and provides collaborative distributed schema-management across distributed sites. The authors do not discuss the need for data pseudonymization or anonymization assuming similarly to [EII+10] that only the members of a research consortium can access the data [WVNL14].

MOSAIC [dlTLAPV13] is a protocol for clinical data exchange with multilateral agreement. MOSAIC was designed to build research databases for private use, and thus, the data privacy is not taken into account in the design of the protocol. Moreover, it is also considered that the different institutions would optionally require more medical data in exchange as queries were made to them. As a result, contributor agents can optionally set a number of medical cases of a certain kind as a requisite, and another (petitioner) agent would have to resolve the requisites imposed by the contributor agents. Using multilateral agreements between agents is proposed by the authors to solve this problem.

Urovi et al. in [UOdlT+14, UOB+12] proposed a secure mechanism for EHR exchange over a P2P agent based coordination framework. In this approach encrypted heterogeneous data are exposed over a P2P network. The authors provide the algorithms for searching and for publishing the EHRs in the untrusted P2P network without compromising the privacy, the integrity and the authenticity of the shared data.

Using unambiguous pseudonym for the patient [Lo 07] enables one to infer additional information about a patient by linking the data from different sources. Xu and Cremers proposed using multiple pseudonyms [XC14], however, their scheme is not applicable for secondary use of medical data. The authors in [NLL07] define two types of pseudonymization: pseudonymization with one-way pseudonyms (which cannot be reversed) and reversible pseudonymization. One-way pseudonyms are not suitable if there is a need to recontact the patient, in case of personalized treatment decisions based on systematic or incidental findings. One disadvantage of the reversible pseudonym is that there is always a risk of de-identification of the individual with corresponding pseudonym. Reversible pseudonymisation may involve one or multiple service providers employed for pseudonymization and de-pseudonymization of the data [PRDS05].

Using encryption combined with pseudonymization techniques [LCHL11, EII+10, XC14] has been recently proposed for building eHealth system in the cloud. There exists, for instance, a number of architectures employing Attribute-Based Encryption (ABE) scheme [YWRL10, IAP09, LYRL10, LYZ+13, LHBC13]. However, these approaches have some limitations, as ABE still can leak information from the access control policy.

Several studies have shown that patients are concerned about their privacy, in particular in the case of medical data sharing: 62% of individuals worry that their electronic health records (EHR) will not remain confidential[13]; 35% expressed privacy concerns regarding the publishing of their data to the database of Genotypes and Phenotypes (dbGaP) [LFS+10]. Therefore, it is unlikely that patients will be willing to share detailed data as this can violate their privacy. One possible approach to address this issue is to guarantee that the shared data are anonymized, but can be traced to ensure that the anonymity is preserved.

As mentioned previously there exist numerous approaches to achieve $k-$***anonymity*** in case of ***local*** (from only one original dataset) and ***static*** (without any updates) data release [FWY05, BA05, KPE+12, LDR05, LDR06, DFMS02]. Usually, to satisfy $k-$***anonymity*** requirements, the generalization and suppression techniques are applied. Data quality and therefore the utility of the dataset is highly dependent on the amount of data available when anonymization algorithm is applied.

In order to enrich anonymized dataset and improve its utility, the following strategies can be applied: ***(i)*** continuously updating anonymized dataset with the new records from the original database, ***dynamic*** data release), and ***(ii)*** using multiple sources of data, or ***distributed*** data release. In the case of incremental update of the anonymized dataset when local database is populated with new records (a), monitoring the difference between multiple versions of the anonymized database may lead to the violation of the $k-$***anonymity*** of the updates (new records). For the dynamic release of stream data, this problem has been addressed by the approaches such as rank swapping [NAT14], modifying quasi-identifiers of already stored records in a privacy preserving way [BSBL06], and accumulating certain amount of records

[13] Health Confidence Survey 2008, Employee Benefit Research Institute

before releasing an update [LOW08]. Furthermore, in [PXW+07] the authors show that privacy leakage is severe in the case of releasing of the anonymized data independently. They propose the monotonic incremental anonymization property and a simple yet effective solution based on $k-$*anonymous* subgroup refinement for maintaining the $k-$*anonymity* against various types of incremental updates in local settings. The authors also conclude that even the subgroup refinement may leak privacy depending on the temporal background knowledge of an adversary. This justifies the necessity of developing new effective methods that provide privacy guarantees in case of incremental updates of anonymized datasets. In [WF06], the authors also address the problem of privacy violation in the case of incremental updates. However, in their framework, the updates are the new attributes added to the schema, not the new records added to the database with the fixed schema.

In case of distributed data release *(ii)*, aggregation of locally anonymized datasets can still reveal sensitive information, especially if the data about an individual are distributed between multiple databases [GKS08, PXW+07]. Several models in the area of distributed privacy-preserving data publishing have already been proposed [EII+10, XC14, CJ05, BLLW11, GKS08]. Ganta et al. [GKS08] were the first to identify privacy breach composition attack: the privacy breach of overlapping population within multiple published datasets. They propose a solution that can be used in the interactive settings (release of statistical information or a query result), but cannot be applied in case of releasing the anonymized data. In [CJ05] the authors describe privacy-preserving algorithm to merge two local $k-$*anonymous* datasets while preserving $k-$*anonymity* property in the resulting dataset. However, the solution is not scalable and requires using secure multi-party computations. Moreover, data-sharing is not independent among different sources contributing to the *RSDB*. An approach for continuous privacy-preserving publishing of data streams is presented in [ZHP+09]. The authors use R-trees, and allow the publication of data into the research database only after performing microaggregation locally. Baig et al. [BLLW11] suggest a new generalization principle, ϵ-cloning, to protect privacy for multiple independent anonymized data-publications. They suggest a model called ϵ-cloning for privacy protection in multiple independent data publications and present an effective algorithm to achieve the ϵ-cloning. The authors in [BLLW11], however, focus on addressing the problems due to the lack of diversity in the sensitive attributes.

However, these models still significantly affect the quality, hence the utility of data, since they do not take into account the availability, content, structure, and representation of the data. In [SS16], the authors discuss the trade-off between privacy and utility of the data, and the risks of breaking anonymity of the data. They state that the risk assessment has to be made for every single situation of data collection. To the best of our knowledge there is no existing work on dynamic improvement of utility of the database in distributed environment with privacy guarantees.

2.6.3 Automatization of TDM Process

In [FCT+13], Fuchs at al. present a review of twelve available clinical pharmacokinetic computer tools. The authors also describe the history and evolution of the software dedicated to monitoring and dosage adjustment staring from the software developed by Laboratory of Applied Pharmacokinetics at the University of Southern California (Los Angeles, CA, USA), launched in 1973 [Jel91] and evolved to the *BestDose*[14], to the available[15] software packages such as *MwPharm* [PM92] and *TCIWorks* [WD11] that turned out to be the best ranked TDM programs according to the review [FCT+13].

MwPharm has the largest database of drugs with their pharmacokinetic properties and almost 300 population models embedded in the software. However, similarly to *BestDose*, it is also a standalone TDM software. This means that the patient's data have to be manually inserted, and dosage adjustments are not automatically sent to the in patient's health record, they will be stored in the local databases. However, according to [FCT+13] none of these programs, including *BestDose* and *TCIWorks*, yet fulfills all of the requirements to clinical pharmokinetics computer program.

During last years (since the review of Fuchs et al. from 2013 [FCT+13]), few TDM software tools have been developed. *TDMx* is a novel web-based open-access support tool for optimizing antimicrobial dosing regimens in clinical routine [WKS+15]. *TDMx* is not a registered or certified medical device. As result of a research project, *TDMx* is provided for personal use only; the accuracy of the provided results can not be guaranteed[16]. Currently, *TDMx* is available for only 4 drugs (*Meropenem, Piperacillin, Amikacin*, and *Gentamicin*).

NextDose is an online dose calculator that uses Bayesian nonparametric approach to propose dose regimens after concentration measurements become available. The software consists of three abstraction layers and provide a clear separation between the user interface, model controllers, and the modeling software. *Doseme*[17] and *insightrx*[18] are recent commercial software tools available for TDM.

However, the authors in [AMM17] claim that currently available software tools (such as *DoseMe*, *insightrx* and *NextDose*) are still sufficiently complex and require training to enable rapid use at the bedside by healthcare professionals.

According to [AMM17], there is still a number of challenges that have to be overcome before individualized drug dosing based on TDM is widely used [AMM17]. Software tools developed to automate the process of TDM are evolving. However, a solution with comprehensive clinical and research capabilities, showing simplicity, flexibility, and user friendliness is still in demand [FCT+13].

[14] http://www.lapk.org/bestdose.php
[15] at the time of preparing the review, year 2013
[16] http://www.tdmx.eu/
[17] https://doseme.com.au/science-behind-doseme
[18] www.insight-rx.com

Privacy in eHealth Part I

3 Secure and Trustable EHR Sharing

Electronic health records (EHRs) contain critical highly-sensitive healthcare information and are frequently shared among peers. Blockchain provides a shared, immutable, and transparent history of all the transactions for building applications with trust, accountability, and transparency. This provides a unique opportunity to develop a secure and trustworthy system by using blockchain for EHR-data management and sharing. In this chapter, we present our perspectives on blockchain-based healthcare data management, in particular, for EHR-data sharing between healthcare providers and for research studies. We propose a framework on managing and sharing EHR data for cancer patient care. In collaboration with Stony Brook University Hospital, we implemented our framework in a prototype by using permissioned blockchain – Hyperledger Fabric v0.6. We provide a security analysis of the proposed framework and discuss the choices of cryptographic solutions required to ensure privacy, security, availability, and fine-grained access control over EHR data. Once adopted by the health community, it will reduce the turnaround time for EHR-data sharing, improve decision making for medical care, and reduce the overall cost.

3.1 Introduction

Electronic medical records contain critical and highly sensitive private information for diagnosis and treatment. EHRs need to be frequently distributed and shared among peers such as healthcare providers, insurance companies, pharmacies, researchers, patients families. This poses a major challenge: keeping a patient's medical history up-to-date. Storing and sharing data between multiple entities, maintaining access control through numerous consents only complicates the process of a patient's treatment. A patient, suffering from a serious medical condition such as cancer, or HIV, has to maintain the long history of his treatment process and post-treatment rehabilitation and monitoring. Having access to a complete history is crucial for the treatment: for instance, knowing the delivered radiation doses or laboratory results is necessary for continuing the anti-cancer treatment.

A patient might visit multiple medical institutions for a consultation or could be transferred

from one hospital to another. According to the legislation, a patient is given a right over his health information and might set the rules and the limits defining who can look at and receive his health information. If a patient needs to transfer his clinical data from one hospital to another or to share his clinical data for research purposes, he will be required to sign a paper-based consent that specifies what type of data will be shared, the information about the recipient, and the period during which the data can be accessed by the recipient. This is difficult to coordinate, especially when a patient moves to another city, region, or country and might not know in advance the caregiver or hospital where he will be receiving care later.

Even if the consent is provided, the process of transferring the data is time-consuming, especially if the hard-copied EHR is sent by post. Most hospital do not send a patient's data via e-mail, because this could impose security risks while a patient's healthcare data are in transit[1]. Ecosystems for health information exchange (HIE), such as CommonWell Health Alliance in the US aim to ensure that the data from EHRs are shared securely, efficiently, and accurately nationwide. This implies that, once providers receive an access to the patient's health information, it is difficult to guarantee that a patient could receive independent opinions from different healthcare providers. Moreover, such ecosystems do not address the requirements in the case of data transferring from one country to another.

Data aggregation for research purposes also requires consent, unless the data are anonymized such that re-identification of a person is not reasonably possible. However, it has been shown that independent release of locally anonymized medical data corresponding to the same patient and originating from different sources (e.g., several healthcare institutions that the patient has visited) could cause re-identification of the patient, hence privacy violations [BLLW11, DUV+15].

Relying on a centralized entity that would solely store and manage patients' data and access-control policies causes having a single point of failure and a bottleneck of the whole framework. This approach also would require either conducting all the operations (such as search, or computations) over encrypted data or choosing a fully trusted entity that has access to sensitive information about the patients. The former still requires management of large amounts of memory [MOO+14] and is not suitable for the hospital environment. The latter was proven to be very difficult to put in practice. The experience with GoogleHealth wallet[2], for example, has shown that patients are concerned about their privacy and aware of the potential risk that their sensitive data might be misused.

Having access to a ledger – shared, immutable, and transparent history of all the actions that happened to all the participants of the network (a patient modifying permissions, a doctor accessing and uploading new data or sharing the data for research) – will overcome the issues presented above. By providing the means to achieve consensus among distributed entities without relying on a single trusted party, blockchain technology will guarantee data security,

[1] http://www.hhs.gov/hipaa/

[2] https://googleblog.blogspot.ch/2009/03/google-health-helping-you-better.html

control of sensitive data, and will facilitate healthcare-data management for the patients and other actors in the medical domain. In the healthcare settings, we can define a transaction as a process of creating, uploading or transferring EHR data; this process is performed within the network of connected peers, e.g., medical institutions. A set of transactions grouped at a certain time is added to the ledger that records all the transactions hence represents the state of the network. The key benefits of applying the blockchain technology in healthcare are the following: a verifiable and immutable history of transactions, tamper resistance, transparency, facilitated access-control management, and integrity of distributed sensitive medical data. This is mainly achieved by employing consensus protocol and cryptographic primitives such as hashing and digital signatures.

The possibility of using blockchain for healthcare-data management has recently raised much attention in the industry[3] and academia [AEVL16, BI16, YWJ+16]. However, only one functioning prototype of a system that uses blockchain for medical-data management has been proposed [AEVL16] to date. In our work, we focus on a practical implementation of a system that uses blockchain technology and that can be integrated into clinical practice. To the best of our knowledge, we are the first to employ permissioned blockchain technology to maintaining metadata and access-control policy and a cloud server to store patients' encrypted data. Combining the blockchain technology and the cryptographic primitives enables us to guarantee data security and privacy, as well as availability with respect to the access-control policy defined by the patient.

The contribution of this chapter is two-fold. First, we propose multiple scenarios of blockchain applications in healthcare and explain our choice of permissioned blockchain technology. Second, we present a framework for blockchain-based data sharing for the primary care of oncology patients under anti-cancer treatment. We then provide a privacy and security analysis of our framework. In collaboration with the department of radiation oncology of a major US hospital, we developed a prototype of this framework. The functionality of the prototype meets the requirements from a medical-practice perspective.

[3] https://gem.co

3.2 Potential Blockchain Applications in Healthcare

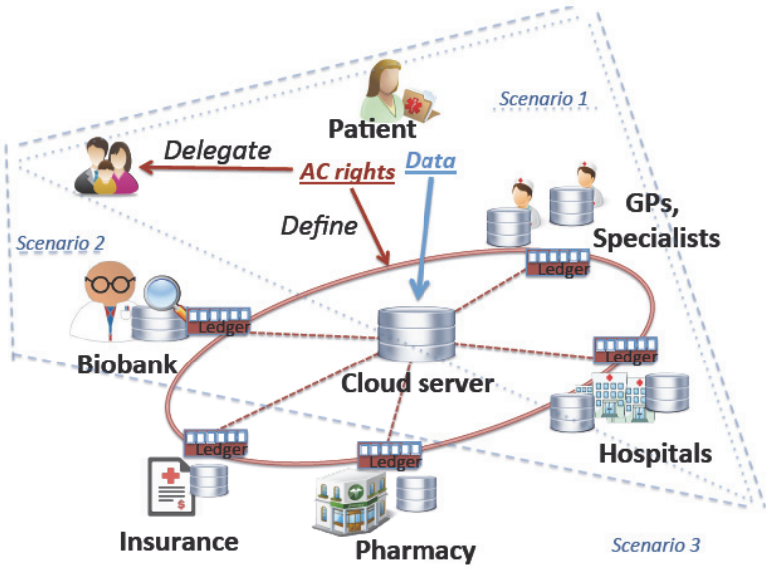

Figure 3.1 – The scenarios of using the blockchain technology in healthcare settings
Scenario 1: Primary patient-care; Scenario 2: Data aggregation for research purposes; Scenario 3: Connecting different healthcare players for a better patient-care.

Blockchain provides a unique opportunity to support healthcare. In this section, we propose three scenarios: primary patient care, medical research, and connected health. Figure 3.1 shows a graphical representation of the combination of the aforementioned scenarios.

Scenario 1: Primary Patient Care. Using the blockchain technology for primary patient care would help us to address the following problems of the current healthcare systems:

- A patient often visits multiple disconnected hospitals. He has to keep his medical history and maintain it up to date. In this situation it might be difficult to ensure that the required information is always available.
- Due to the unavailability of the data, a patient might have to repeat some tests for laboratory results. This is common when the results are stored in another hospital and cannot be immediately accessed.
- The healthcare data are sensitive and the data management is cumbersome. Yet, often, there is no privacy-preserving system in clinical practice that enables patients to maintain an access-control policy in an efficient manner.
- Sharing data between different healthcare providers could be time consuming and require major efforts from the patients and medical professionals.

Next, we propose two approaches that can be implemented separately or combined to improve patient care.

- *Institution-based*: The network would be formed by trusted peers: healthcare institutions or general practitioners (caregivers). The peers will run a consensus protocol and maintain a distributed ledger. A patient (or his relatives) will be able to access and manage his data through an application at any node where his information is stored. If a peer is off-line, a patient could access the data through any other online peer. The key-management process and the access-control policy will be encoded in a chaincode, thus ensuring data security and the patient's privacy.
- *Case specific* (serious medical conditions, examination, elderly care): During a patient's stay in a hospital for treatment, rehabilitation, examination, or surgery, a case-specific ledger could be created. To achieve efficiency and transparency of the treatment the network would connect doctors, nurses, and family members, and it would ensure that all the required data are available to the healthcare professionals at all the stages of the treatment. This might help to decrease the number of human-made mistakes.

Scenario 2: Data Aggregation for Research Purposes. It is important to ensure that the sources of the data are trusted medical institutions and that the data are authentic. Using a distributed ledger will guarantee traceability and patients' privacy, as well as the transparency of the data aggregation process. Due to the current lack of appropriate mechanisms, patients are often unwilling to permit data sharing. Using blockchain technology within a network of healthcare professionals, researchers, biobanks, and healthcare institutions would facilitate the process of collecting medical data for research purposes.

Scenario 3: Connecting Different Healthcare Players for Better Patient Care. Connected health is a model for healthcare delivery that aims to maximize healthcare resources and to provide opportunities for consumers to engage with caregivers and improve self-management of a health condition. Sharing the ledger (using the permission-based approach) among entities (such as insurance companies and pharmacies) will facilitate medication and cost management for a patient, especially in case of chronic disease-management. Providing pharmacies with accurately updated data about prescriptions will improve the logistics. Access to a common ledger would enable the transparency in the whole process of the treatment, from monitoring if a patient follows correctly the prescribed treatment to facilitating communication with an insurance company that regards the costs of the treatment and medications.

Implementing the Scenarios. In order to implement the healthcare scenarios presented above, we must choose between an existing to-date permissionless implementation and permissioned blockchain one. Next, we present the facts that favor a permissioned implementation over a permissionless one.

- The anonymity of users and impossibility to verify the identity of account holders (as in

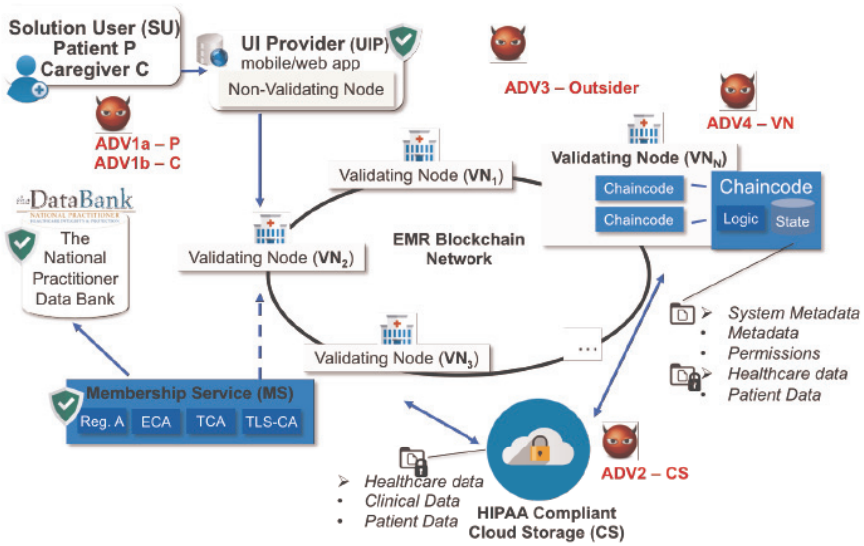

Figure 3.2 – The system model for blockchain-based EHR-data sharing using Hyperledger Fabric v0.6

the case of permissionless blockchain) could cause data misuse.

- Patients' healthcare data are of a highly sensitive nature. In the case of a permissionless blockchain, monitoring the communications between a patient and a specific clinician could reveal some sensitive data about the patient, hence violating his privacy.
- The rapid response of a system is required, as every update of the information about a patient's treatment could be crucial for the patient.
- The need to pay for an execution of transactions: For example, updating permissions for a medical doctor to access a piece of healthcare information or sharing some data for research could limit the usability of the system.

3.3 System Model

In this section, we introduce a system model by defining the components of the system and the communications between them. Next, we define the threat model by discussing the roles and the possible behaviors of the components of the system, making the assumptions and defining the design goals accordingly.

System Model

Figure 3.2 shows the system model for blockchain-based EHR-data sharing using Hyperledger Fabric v0.6. The system consists of the following components.

Validating nodes (VNs) are the nodes represented by the medical institutions that form the EHR blockchain-network. The chaincode is deployed at every *VN* for the management of the

healthcare data provided by the patient and the system metadata: permissions and healthcare metadata. Validating peers agree on the validity of the transactions submitted by the users and update the blockchain by adding new blocks. *UI Provider (UIP)* is deployed within clinical infrastructure and plays the role of a non-validating node (constructing and forwarding transactions from the users to *VN*s).

Solution user (SU) is a user of the system with the following currently available roles: caregiver *C* (a medical doctor from a hospital, or an independent medical practitioner) and patient *P*. *SU*s are registered at membership service, *MS*, and could interact with the blockchain, according to their roles through the interface of a web-application (*UIP*). The users are able to generate secret keys and passwords. The widespread use of the electronic identity cards and the cards provided by the insurance companies shows that having a smart card is not a burden in everyday life. Therefore, we assume that it is possible for an *SU* to use a smart card. The keys could also be stored on the mobile phone with the chip. We assume that an *SU* patient, *P*, is able to express his access-control policy decide with whom to share the access to different types of the healthcare data.

Membership service (MS) is an entity that manages the network identities of all member organizations and users. *VN*s are registered at the *MS* that uses a trusted database (such as the National Practitioner Data Bank) when enrolling a new *VN*. We assume that the membership service hosts a standard certification authority (*CA*) that can generate certificates for a key pair for signing (SK^S_{SU}, PK^S_{SU}) and an encryption key pair (SK^E_{SU}, PK^E_{SU}) for every user (*SU*), and corresponding key pairs for *VN*s.

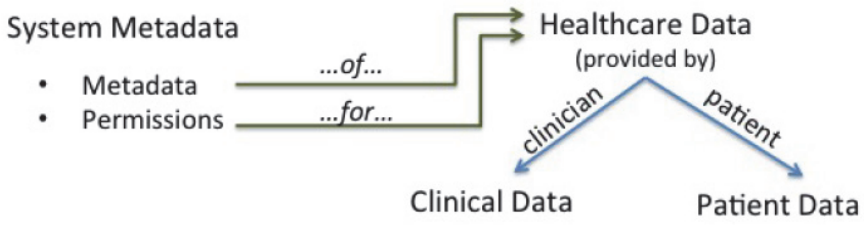

Figure 3.3 – The healthcare-data structure

HIPAA compliant cloud storage (CS) is a HIPAA-compliant cloud server where highly sensitive healthcare data are stored in the encrypted form temporary (according to the access-control policy specified by the patient). As shown in Figure 3.3, the healthcare data can be provided by the patient or by his clinician. System metadata consists of the metadata (e.g., data source, data category) and the permissions corresponding to each healthcare-data item.

System Assumptions and Threat Model

MS manages the identities of *VN*s and *SU*s but does not have access to to the healthcare data or the system metadata stored on the blockchain. In the current implementation, we assume that the membership service is trusted and cannot be compromised by the adversary[4].

We restrict our work by following the settings of the permissioned blockchain technology (we explain our choice in the previous section). In the current prototype implementation, we also use a single certification authority. However, alternative implementations are envisaged for the subsequesnt releases of Hyperledger Fabric, such as support of anonymous credentials with multiple certification authorities and the use of threshold signatures.

We assume that a UI Provider, *UIP*, is a trusted software deployed within a clinical infrastructure to construct and forward transactions from *SU* to *VN*. To ensure that the software is trustworthy, the source code can be digitally signed and be made available as an open source for the verifications.

We assume that secure communication channels have been established over HTTPS between all parties in the system. We assume that strictly less than a third of *VN* can collude, and no collision is possible between caregivers, membership service, and a cloud server. We further assume that the data from the network layers (e.g., IP address) cannot be used to leak users' identities. This assumption is reasonable, as many users only access the Internet through a NAT gateway offered by their Internet provider, and could be relaxed if, for instance, users and validating nodes employ a VPN service or anonymous networks (e.g., Tor) to access the Internet.

In our model, we assume that the adversaries can be among solution users (**ADV1**), a cloud server (**ADV2**), an outsider, any user that is not registered in the system (**ADV3**), and validating nodes (**ADV4**).

ADV1: Patient (**ADV1a**) can be an active adversary. For instance, a malicious user could try to impersonate another user and to gain access to another patient's healthcare data, or to learn the system metadata (e.g., permissions set up by the patient for the caregivers). We assume that caregiver (**ADV1b**) is honest-but-curiuos, and will not collude with other caregivers: *C* will use the system to verify his rights, to read the data (provided by the other *C*s, and *P*s) and upload new healthcare data about the patient according to the patient consent. We do not try to protect against false information provided by *C*, as we assume that medical doctors do not have incentive for generating false content. We assume that *C* is not aware of therefore will not be able to follow the access control policy specified by the patient regarding the healthcare data uploaded by other caregivers and the patient before *C* uses our system.

ADV2: A cloud server can be an honest-but-curiuos adversary. It is employed to temporily store shared patient's data. *CS* is a HIPAA-compliant cloud storage, i.e., it manages the protected health information (PHI) and ensures that all the required physical, network, and process

[4]To relax the assumptions and to provide stronger security and distribute trust single MS could be substituted by Collective Authority servers. An example of a scalable solution is presented in [STV+16]

security measures are in place and followed. We assume that *CS* will not collude with any other entity. *CS* will store the encrypted data and provide an access, only according to the access control policy stored on the blockchain (including the deletion of the patient's data after access-control policy is expired). However, *CS* might try to gain access to the highly sensitive healthcare data.

ADV3: Outsiders, or external observers, could try to get access to the healthcare data, system metadata, or cryptographic keys by trying to impersonate solution users, or to create fake profiles in the system.

ADV4: *VN*s are modeled as Byzantine nodes. To ensure consensus among the nodes maintaining the blockchain, among $3f+1$ nodes the maximum number of faulty nodes is f. Therefore, we assume that strictly less than a third of *VN*s can be malicious or/and collude. Given the application scenario (presented in Section 3.4.1) and the medical context, we assume that medical institutions are committed to providing compliant data-sharing for better patient-care and will not have incentive to behave maliciously but can still be temporarily unavailable. We assume that *VN* will follow the steps required to reach consensus and will maintain the blockchain to ensure that patient's access-control policy is expressed on the ledger and can be accessed by the authorized users.

Design goals
We define the following functionality for the registered *SU*s: patient P (F^P) and caregiver C (F^C), respectively.

- $F^P(i)$: Patient P can create a new record on the blockchain to manage his permissions, the metadata of the information stored on the *CS*, and some of his healthcare data.
- $F^P(ii)$: Patient P can access his record by querying validating nodes (by sending `query` transaction).
- $F^P(iii)$: Patient P can add permissions that correspond to the certain type of data uploaded by the caregivers or by the patient himself. These permissions grant the right to caregiver C to upload, read, and share the data about patient P (`invoke` transaction).
- $F^P(iv)$: Patient P can upload to *CS*, and to the blockchain, different kinds of healthcare data.
- $F^C(i)$: Caregiver C can query the ledger to view the permissions specified by the patients.
- $F^C(ii)$: Caregiver C can request access to the data from *CS*; this will require verification of the permissions by querying the validating nodes.
- $F^C(iii)$: Caregiver C can upload the healthcare data about the patient to *CS* if there is a corresponding permission specified by the patient.

We provide the aforementioned functionality and ensure the data security, the patient's privacy, and traceability of all the actions of *SU*s, with respect to the shared data. In particular, for the highly sensitive healthcare data and the system metadata, which can also be sensitive, our design goals are the following.
Data integrity: the solution users are ensured that the data were not altered in transit or at rest by an adversary.
Authenticity: the users are ensured that the data were sent by the claimed sender.
Availability: the data are available from anywhere at any time, according to the access-control policy specified by the patient.
Confidentiality: the information disclosure to unauthorized individuals is not possible.
Unlinkability between system metadata and the corresponding patient's identity for any adversary: only the users authorized by the patient are permitted to link the patient's identity and his record stored on the blockchain.

3.4 Application in Radiation Oncology: Sharing Clinical Data between Healthcare Providers

In this Section, we present a working prototype: design and implementation of a system to support EHR-sharing for primary patient-care (Scenario 1). More precisely, we focus on patients that receives an anti-cancer treatment via ionizing radiation, usually performed in the department of radiation oncology of a hospital.

3.4.1 System Overview

We use the blockchain technology to create a prototype of an oncology-specific clinical-data sharing system. Our solution facilitates the consent management and accelerates data transfer in a privacy-preserving way. We developed a chaincode that enables a patient to easily ensure his fine-grained access-control policy for his data and that enables efficient data sharing among clinicians.

To present our solution, we take as an example an oncology information system (e.g., ARIA[5]) widely used to facilitate oncology-specific comprehensive information and image management. Such systems combine radiation, medical, and surgical oncology information and assist clinicians to manage different kinds of medical data, to develop oncology-specific care plans, and to monitor radiation doses for patients. Different types of data are stored in these systems and need to be structured and exported, depending on the clinician's request. The documents that contain these data (history and physical exams, laboratory results, and delivered radiation doses) are of the high importance for the clinicians.

We assume that the EHR blockchain network is formed by the medical institutions that play

[5]https://www.varian.com/oncology

the role of validating nodes. The web applications (*UIP*s) are deployed in the protected environment (such as the infrastructure of a hospital) and can be used by the registered solution users. The chaincode is deployed at all the validating nodes. The cloud server is set up and is HIPAA-compliant.

Solution users, as well as validating nodes, are registered in the network (identity of every caregiver is verified using the National Practitioner Data Bank) with their credentials: an ID and a password. The *CA* is used to generate the certificates for the public keys of the following key pairs: for signing $(\mathsf{SK}^{\mathsf{S}}{}_{SU}, \mathsf{PK}^{\mathsf{S}}{}_{SU})$ and encryption of the off-chain communications $(\mathsf{SK}^{\mathsf{E}}{}_{SU}, \mathsf{PK}^{\mathsf{E}}{}_{SU})$ for the users, and a key pair for signing the transactions $(\mathsf{SK}^{\mathsf{S}}{}_{VN}, \mathsf{PK}^{\mathsf{S}}{}_{VN})$ for validating nodes. *P* also generates the secret keys to be used to generate his pseudonym, to encrypt the data stored on the cloud $(\mathsf{SK}^{\mathsf{symm}}{}_{P})$, and the key $(\mathsf{SK}^{\mathsf{symm,CC}}{}_{P})$ to encrypt his healthcare data to be stored on the blockchain in case of emergency (for the latter, $\mathsf{SK}^{\mathsf{symm,CC}}{}_{P}$ can be shared, for instance with the patient's relatives, who are registered in the system as another patients).

We assume that the hospitals can provide an infrastructure to securely store the sensitive data and the cryptographic keys of *VN*s and the caregivers. Managing the credentials and keys of the patients and independent caregivers could be implemented using two-factor authentication – smart cards and passwords – to provide stronger security guarantees.

After the registration, the user generates a pseudonym and invokes the transaction to create his record on the blockchain. Before any transaction occurs, a user is authorized with the credentials generated during the registration process. A chaincode is designed such that, before the record is created on the blockchain, it is verified whether a record corresponding to this user already exists. Nothing prevents a user from creating multiple symmetric keys $\{\mathsf{SK}^{\mathsf{symm}}{}_{P_i}\}$ and for generating multiple records. However, the healthcare-data management could become more complicated for the patient. For simplicity, we assume that a patient will only generate another pseudonym and a new record if his currently used key $\mathsf{SK}^{\mathsf{symm}}{}_{P}$ is compromised or lost. We discuss the case of multiple records that belong to the same patient and the case of lost or compromised $\mathsf{SK}^{\mathsf{symm}}{}_{P}$, in Section 3.5.

We define the two types of the data, as presented in Figure 3.3. The first one is the sensitive healthcare data corresponding to the patient. In our prototype, we use the following data categories: history and physical exams, laboratory results, and delivered radiation doses. Different data categories could be defined based on the semantics and the sensitivity level of the data. The second type is the system metadata: the metadata of healthcare data and permissions, specified by the patients. Patients and clinicians upload the healthcare data to the *CS* by using *UIP*. The *SU*s also define the parameters required to construct the transactions that will update metadata on the blockchain by using *UIP*.

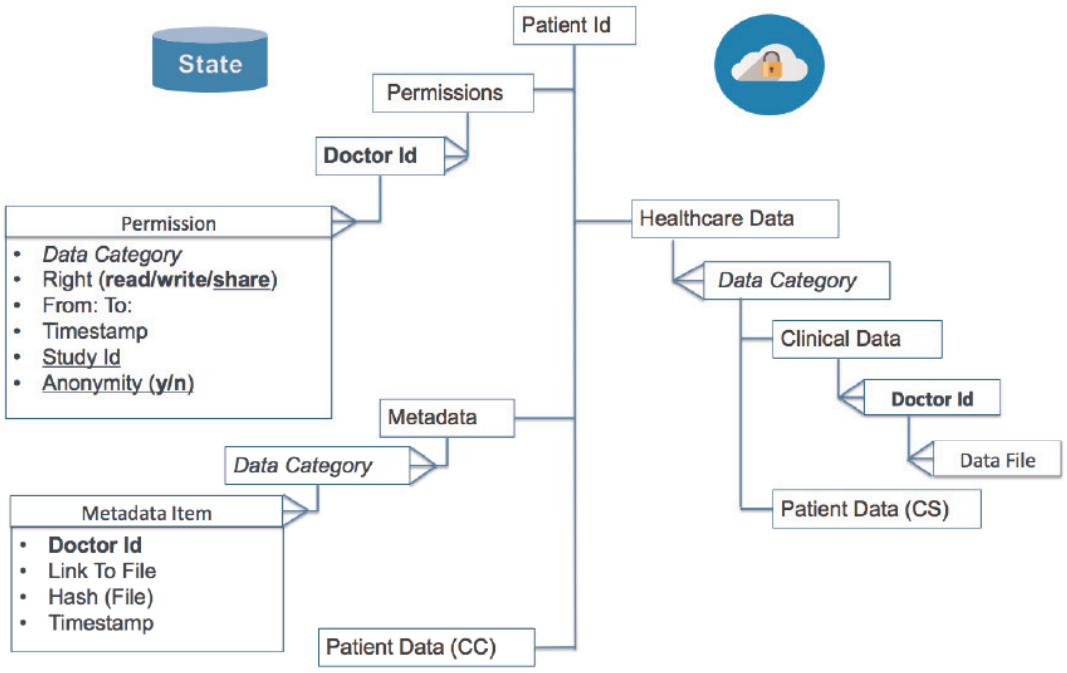

Figure 3.4 – The structure of the data stored on the chaincode and the cloud server

Figure 3.4 presents the detailed structure of healthcare data and system metadata that are stored on the chaincode and the cloud server.

The block containing the information about the permissions is organized as follows. Every permission corresponds to an ID with which a clinician is registered in the system. Every permission specifies the timeframe (`from: to:`), during which the clinician has a `right` to **read** the patient's data that fall into a specific `data category`, to upload the data to the cloud repository (**write**), or to **share** the patient's data within a framework of a specific research study, `study id`. For the latter, the patient could also use an `anonymity` tag to specify if the data must be anonymized before sharing or could be shared as they are. `Timestamp` enables the patient to update and track access control changes.

For patient P to revoke the right for caregiver C to access a specific type of data provided by other caregivers, P has to add a new permission with another time frame.

Clinical metadata is a block that contains information about all the data files uploaded to the cloud by the clinicians or the patient himself. The metadata items are categorized based on the semantics of the corresponding data files. Every item contains an ID of the clinician that uploaded the data (`doctor ID`) or a patient's pseudonym, a pointer to the file that is stored in the cloud (`path to file`) the hash of the data file (`hash(file)`) to ensure unforgeability of the data stored in the cloud, and the `timestamp` of the moment when the data file was uploaded. Similarly, a patient could upload some of his healthcare information to the

chaincode.

```
{"patientId": "pid1",
"privateData": "some data of the patient",
"data": [
  {
    "categoryId": "Lab Results",
    "dataEntry": [
        "dataDoctorId": "doc1",
        "dataLink": "https://.../file01.pdf",
        "dataTimestamp": "2016-11-28 20:35:18 0000 UTC",
        "hashOfFile": "7217822155b9aea53...4a71150ded22d9"
     } ]
  },
  {
    "categoryId": "Radiation Dose",
    "dataEntry": [...]
  }
],
"permissions": [
  {
    "doctorId": "doc1",
    "permissionDoc": [ {
        "right": "write",
        "categoryIdP": "Lab Results",
        "TimestampP": "2016-11-28 20:35:18.595854445 0000 UTC",
        "TimeFrom": "Mon Nov 21 15:04:05 MST 2016",
        "TimeTo": "Mon Dec 5 15:04:05 MST 2016"
    } ]
  }
]}
```

Figure 3.5 – Example of the patient's record that contains metadata and permissions in JSON format

Listing 3.5 presents an example of the metadata stored as a byte array as a "value" part of a CC State. "Value" is associated with a "key" – a pseudonym of the corresponding patient. The patient's pseudonym is generated by applying a hash function on the concatenation of the patient symmetric key $\mathsf{SK}^{\mathsf{symm}}{}_P$, and a uniquely identifiable information of the patient (UII_P): $\mathsf{H}(\mathsf{SK}^{\mathsf{symm}}{}_P \| \mathrm{UII}_P)$. The combination of the following information could be used as UII_P: social security number, SSN, (if applicable), date of birth, given names, and the ZIP code of the patient.

3.4.2 Healthcare-Data Sharing

We first provide the description of the data-sharing protocol. Then, we provide the formal definition of the cryptographic interfaces that we propose to ensure that the design goals defined in Section 3.3 are met. Finally, we describe the prototype implementation of the chaincode and discuss its scalability.

Data-Sharing Protocol.
The protocol for the data sharing can be described with the following steps.

Step 1. *P uploads/downloads his data to/from the system.*

- **Step 1(a).** Patient *P* encrypts his data to be stored on the blockchain (patient data (CC)) and invokes the transaction to update his record on the blockchain.
- **Step 1(b).** Patient *P* uploads his encrypted data and updates his record on the blockchain by adding the metadata information (metadata item of the patient record stored on the blockchain) about the corresponding data uploaded to the cloud. These information includes the path to the file, hash of the data, and the timestamp.
- **Step 1(c).** Patient *P* can download his data that have been uploaded earlier by himself or clinical data provided by the caregivers (as described in the next steps) if the data are still stored on the *CS* (according to the duration of the permissions specified by *P*). For this, the patient needs to log in at the mobile (web) application, then he can query the metadata stored on the blockchain to see the data that are currently available and can be downloaded from *CS*.

Step 2. *P shares his key.* To share the data with caregiver *C*, patient *P* needs to share his key $\mathsf{SK}^{\mathsf{symm}}{}_P$ with *C*. This step is done off-chain, using the public key of caregiver *C*. Once the caregiver has the patient's key, $\mathsf{SK}^{\mathsf{symm}}{}_P$, *C* can generate the pseudonym and query the patient's record on the blockchain.

Step 3. *P adds a permission.* Patient *P* invokes the transaction to add a permission that allows the registered caregiver to read/share the data that are already uploaded to *CS* and to write (add) the data to *CS*. Permissions are also indirectly used to delete the data corresponding to the patient *P* from the cloud.

Step 4. *Permissions verification.* Caregiver *C* verifies if he can access the data of patient *P*. For this, *C* queries a validating node. When the *SU* caregiver queries the ledger (functionality $F^C(i)$), the chaincode implementation ensures that *C* only learns about the permissions specified by *P* for *C*. If *C* already has the patient's key $\mathsf{SK}^{\mathsf{symm}}{}_P$, *C* can send a query to *CS* (**Step 5**). Otherwise, *C* has to request patient *P* to share his key $\mathsf{SK}^{\mathsf{symm}}{}_P$ off-chain (**Step 2**).

Step 5. *C reads/writes/shares the data.* Caregiver sends a query to the *CS* in order to access or upload the healthcare data for the patient's treatment, or in order to share the data for research purposes. At this step, permissions specified by the patient are enforced by setting up a threshold: the *CS* has to obtain signed confirmations from at least two-thirds of the *VN*s running the chaincode. When the clinician adds new healthcare data, a non-validating node constructs the transaction that will update the patient's record on the blockchain by adding a *metadata item* to the corresponding patient's record with the following information: the ID of caregiver *C*, the path to the file, the hash of the data, and the timestamp. The patient

triggers the uploading of the healthcare data by sending a request sent to a caregiver via a web application. However, the data can be uploaded only if there are corresponding permissions specified by the patient on the blockchain.

Step 6. *P triggers deletion of the data stored on the CS.* A patient has the right to delete his data from the *CS*. For this, *P* sends a request to the *CS* and if there are corresponding permissions for this category of the data, the data are erased. The verification is analogous to the verification process of the **Step 5**.

Cryptographic Interfaces

Figure 3.6 provides the formalization of the cryptographic interfaces available for the users (*SU*): patients and caregivers.

For the solution user *SU*:

- $(\mathsf{SK}^{\mathsf{S}}{}_{SU}, \mathsf{PK}^{\mathsf{S}}{}_{SU}) \leftarrow \mathsf{Sig.KeyGen}(1^{\lambda}), (\mathsf{SK}^{\mathsf{E}}{}_{SU}, \mathsf{PK}^{\mathsf{E}}{}_{SU}) \leftarrow \mathsf{Enc.KeyGen}(1^{\lambda})$ - generation of the public and private keys for signing and encryption for solution user *SU*;
- $(trans \| s_{trans} \leftarrow \mathsf{Sig.KeyGen}(\mathsf{SK}^{\mathsf{S}}{}_{SU}, trans))$ - transaction signing for the interactions with CC when exploiting the functionality of the chaincode ($F^{P}(i) - F^{P}(iv)$ for patient *P*, $F^{C}(i) - F^{C}(ii)$ for caregiver *C*) ;
- for *SU* that has $\mathsf{SK}^{\mathsf{S}}{}_{P}$:
  - $PS_P \leftarrow \mathsf{H}(\mathsf{SK}^{\mathsf{symm}}{}_{P} \| \mathrm{UII}_P)$ - a pseudonym generation for a patient *P* (for *SU* that has $\mathsf{SK}^{\mathsf{S}}{}_{P}$) ;
  - $c_{Data} \leftarrow \mathsf{Enc}^{S}.\mathsf{Encrypt}(Data, \mathsf{SK}^{\mathsf{S}}{}_{P}), h_{Data} \leftarrow \mathsf{H}(c_{Data}), (c_{Data} \| h_{Data})$ - uploading the data on the *CS* (according to the permissions of P) ;
  - $Data \leftarrow \mathsf{Enc}^{S}.\mathsf{Decrypt}(c_{Data}, \mathsf{SK}^{\mathsf{symm}}{}_{P}) \iff h_{Data} \leftarrow \mathsf{H}(c_{Data})$ - decryption and verification of the data related to patient *P* and stored on the *CS*.

Specific for patient P:

- $\mathsf{SK}^{\mathsf{symm}}{}_{P} \leftarrow \mathsf{Enc}^{S}.\mathsf{KeyGen}(1^{\lambda})$ - generation of a patient-specific symmetric key ;
- $c_{\mathsf{SK}^{\mathsf{symm}}{}_{P}} \leftarrow \mathsf{Enc.Encrypt}(\mathsf{SK}^{\mathsf{symm}}{}_{P}, \mathsf{PK}^{\mathsf{E}}{}_{C}), s_{\mathsf{SK}^{\mathsf{symm}}{}_{P}} \leftarrow \mathsf{Sig.Sign}(c_{\mathsf{SK}^{\mathsf{symm}}{}_{P}}), \mathsf{SK}^{\mathsf{S}}{}_{P}),$ $(c_{\mathsf{SK}^{\mathsf{symm}}{}_{P}} \| s_{\mathsf{SK}^{\mathsf{symm}}{}_{P}})$ - sharing the patient's specific key with caregiver C.

Specific for caregiver C:

- $\mathsf{SK}^{\mathsf{symm}}{}_{P} \leftarrow \mathsf{Enc.Decrypt}(c_{\mathsf{SK}^{\mathsf{symm}}{}_{P}}, \mathsf{SK}^{\mathsf{E}}{}_{C}) \iff 1 \leftarrow \mathsf{Sig.Verify}(c_{\mathsf{SK}^{\mathsf{symm}}{}_{P}} \| s_{\mathsf{SK}^{\mathsf{symm}}{}_{P}},$ $\mathsf{PK}^{\mathsf{S}}{}_{P})$ - decryption and verification of the authenticity of the symmetric key of patient P.

Figure 3.6 – Functionality of the interfaces for *SU*s

All the solution users possess two different key pairs for signing and encryption. The keys are generated during the registration phase. With his secret key for signing ($\mathsf{SK}^{\mathsf{S}}{}_{SU}$), the *SU* signs every transaction when exploiting the functionality of CC.

A user that possesses the symmetric key of patient *P*, $\mathsf{SK}^{\mathsf{symm}}{}_{P}$, (i.e., patient *P* and the caregivers with whom *P* has shared his key) could also generate the pseudonym of the patient in order to query the patient's record on the blockchain. $\mathsf{SK}^{\mathsf{symm}}{}_{P}$ is also used to encrypt the data before uploading the data to the *CS* and to decrypt the patient's data. As mentioned before, the patient could store his data on the *CS* and some limited amount of the data on the

blockchain. For each type of the data, different symmetric key is generated and used similarly to the $SK^{symm}{}_P$, as described above.

A patient can generate multiple symmetric keys for different kinds of data and share the keys with a caregiver (and/or relatives). The keys are shared off-chain. The public-key encryption of the keys ensures confidentiality, and the digital signature ensures integrity and authenticity. Caregiver *C*, in turn, can verify the authenticity of the ciphertext generated by the patient using the patient's public key; C, using his private key, can decrypt the patient's symmetric key.

Chaincode Prototype and its Scalability

To test the functionality of the developed chaincode, we built a network that consists of a membership service and four *VN*s executing PBFT consensus protocol. We developed the chaincode that provides the functionality described above. We deployed the chaincode on every *VN* in our network and tested the system functionalities that can be provided for both patients and caregivers (F^P, and F^C as defined in Section 3.3). We tested it by creating of a new patient record on the blockchain, by adding a permission, a metadata, and a data item, by querying the chaincode, and by verifying the permissions.

For instance, before a clinician is able to add new data to a cloud repository, a permission corresponding to this clinician is retrieved from the patient's metadata record. Then, the validity of the permission, with respect to the data category and the timeframe, is verified. Similarly, sharing non-anonymized patient data for the research purposes cannot be performed by the clinician without the patient's agreement.

We refer the reader to the Section 3.5 for the security analysis of the proposed architecture and the discussion about the choice of cryptographic solutions required to ensure security and privacy. The system will be interfaced with the existing clinical database-management systems, and more experiments with the data of the real patients will be conducted.

Clinical-data sharing requires the scalability of the system, in terms of both the number of users and the number of nodes. PBFT consensus protocol provides excellent scalability in terms of the number of users, but has not been well explored in terms of the number of nodes (verified only up to few tens of Nodes) [Vuk15]. The scalability issues could be addressed by using hierarchical BFT protocols. The frequency of creating a block or number of transactions in a block (batch size) could be also adjusted. A system load is already minimized by storing off-chain patient's clinical records. Next, we discuss the privacy and security of the proposed framework.

3.5 Privacy and Security Analysis

Hereafter, we present the privacy and security analysis of the data-sharing protocol described in Section 3.4.2 in order to show that the privacy and security design goals defined in Section 3.3 are satisfied.

Data integrity and authenticity. To ensure data integrity and authenticity, we have to guarantee that the *SU*s are ensured that the data were not altered in transit or at rest by an adversary and that the data were sent by the claimed sender.

The integrity and authenticity of the *healthcare data* can potentially be violated at the following steps of the healthcare data-sharing protocol: *(i)* when *SU*s upload the data to the *CS* (**Step 1(b)**: *P* uploads his data, **Step 5**: *C* shares the clinical data); *(ii)* during the time that the data are being stored on the *CS* (from the moment they were uploaded by an *SU* till the expiration of the corresponding permissions); and *(iii)* when the data are being downloaded from *CS*.

The integrity and authenticity of the *system metadata* can potentially be violated every time a transaction is issued by a solution user or the *CS* and when the validating nodes run consensus protocol to update the ledger. In particular, in the protocol presented above, the integrity and authenticity of the system metadata can be at risk in the following cases: when patient *P* updates or queries his record on the ledger (**Step 1(b)**), when the caregivers verify their permissions on the ledger (**Step 4**), when *CS* queries the validating nodes (**Step 5**), and from the moment system is running and the validating nodes must maintain the ledger.

The threats to the integrity and the authenticity of the *cryptographic keys* exist during the whole lifecycle of the private keys. Moreover, the integrity and the authenticity of the keys can be violated when patient *P* wants to share his data and secret keys ($\mathsf{SK}^{\mathsf{symm,CC}}{}_P$ and $\mathsf{SK}^{\mathsf{symm}}{}_P$) with caregivers.

To ensure the authenticity of the healthcare data and the system metadata provided by the *SU*s, the patients and the caregivers digitally sign all the transactions with their corresponding private keys ($\mathsf{SK}^{\mathsf{S}}{}_{SU}$). To guarantee the authenticity and the integrity of the healthcare data, system metadata are stored on the blockchain. System metadata consists of the hash of the corresponding healthcare-data file uploaded to the cloud server and the information about the user that uploaded the data file. Every transaction initiated by the users and forwarded to a *VN* by a *UIP* is signed by the private key of validating node, $\mathsf{SK}^{\mathsf{S}}{}_{VN}$. Before a transaction can be forwarded to a *VN*, it is constructed by a non-validating node that can be accessed only by the authorized users. When the *CS* or *SU* queries the ledger, the replies from the validating nodes are signed to ensure data authenticity and integrity. Because we assume that the validating nodes and the solution users can securely manage the secret keys and the credentials generated during the enrollment, the *integrity* and *authenticity* of the healthcare data and the system metadata are guaranteed by the correctness and unforgeability of the digital signature algorithm and the properties of a secure cryptographic hash function (listed in Section 2.2).

The integrity of a patient's record stored on the blockchain is based on the employed consensus protocol that ensures an atomic broadcast and its properties: validity, agreement, integrity, and total order.

The secret keys of the patients and independent caregivers are stored on the smart card or the mobile phone protected by the pin code known only to its owner. Therefore, no adversary can access and/or tamper the keys at rest. Patients also sign their encrypted secret keys ($SK^{symm,CC}{}_P$ and $SK^{symm}{}_P$) when sharing them with the caregivers (**Step 2**). Caregivers store their secret keys and shared keys with the patients in the secured clinical infrastructure. The keys can only be accessed after a two-factor authentication (e.g., a badge of the medical doctor and his password, received at the registration). If private keys $SK^{S}{}_{SU}/SK^{E}{}_{SU}$ of a user are lost or compromised by an adversary, the fresh keys and certificates will be signed by a membership service, based on the uniquely identifiable information of the *SU*.

Availability guarantees that the data can be accessed from anywhere at any time. However, in our system, we require availability of *healthcare data*, *system metadata*, and *cryptographic keys* with respect to the access-control policy specified by the patient. The availability of *healthcare data* can be at risk if the *CS* is off-line, if the data were deleted or never uploaded to *CS*, if there is no corresponding permission, and if a user does not possess the corresponding secret key to decrypt the data. The availability of *system metadata* cannot be guaranteed if there is one third or more malicious or off-line validating nodes or if all the non-validating nodes (web applications) are down. The availability of *cryptographic keys* is at risk if the *SU* loses his credentials and/or his smart card.

As per the assumptions listed in Section 3.3, the healthcare data are stored on a HIPAA compliant *CS* (that also provides backup services) with respect to the permissions specified by the patients. The data are erased when the permissions stored on the ledger are expired (**Step 6.**). We also assume (in Section 3.3) that the cloud is an honest-but-curious adversary, hence it will not erase the data unless the corresponding permissions are expired. Creating the permissions and sharing the keys are the patient's responsibility, and we assume that the patient is able to express his consents/access-control policy by specifying the permissions and sharing their keys, respectively, through the web interface.

The system metadata are available for the authorized users via the blockchain; the availability is guaranteed by the properties of the atomic broadcast that is ensured by employing PBFT consensus protocol when there is strictly less then one third of faulty validating-nodes (**ADV3**). We also assume that web applications (*UIP*s) are trustworthy and are available in multiple hospitals.

To ensure the availability of the healthcare data, the cryptographic keys are managed in the following way. Different symmetric keys are generated for the encryption of the patient's healthcare data that are stored on the blockchain ($SK^{symm,CC}{}_P$) and the *CS* ($SK^{symm}{}_P$). If a symmetric key $SK^{symm}{}_P$ is lost, a new symmetric key $SK^{symm'}{}_P$ can be generated by the patient. The data stored on the *CS* have to be deleted by modifying the permissions (**Step 3**) and triggering the deletion of the patient's data stored on the *CS* (**Step 6.**). A new pseudonym and a new patient's record will be created on the blockchain. Then, the patient will have to add the permissions and request the caregivers to upload the patient's healthcare data encrypted

with $SK^{symm'}_{P}$ to *CS*. The patient can request his pseudonym by using *UIP* from one of the caregivers with whom he shared $SK^{symm}{}_{P}$. This gives him an opportunity to re-generate his pseudonym and, due to the immutability property of the blockchain, to access his old record. Nothing prevents the patient from generating multiple keys $\{SK^{symm}{}_{P}\}$ and creating multiple records. However, the multiple records being stored on the ledger and corresponding to the same patient cannot provide better privacy for the patient; multiple records can be linked to the same user, but due to the properties of a secure hash function, this will not help to re-identify the patient who is using multiple pseudonyms. Moreover, having multiple records corresponding to the same patient could complicate ensuring data availability and implies that the patient will have to manage multiple records and secret keys.

Confidentiality is ensured when the disclosure of information to an unauthorized individual is not possible. In our system architecture, we ensure the confidentiality of the *healthcare data*, *system metadata*, and *cryptographic keys*. The confidentiality of the *healthcare data* could potentially be violated from the moment the data are sent from the local database of the medical institution/caregiver/patient to the *CS*. The confidentiality of the data can be violated when the data are in transit, i.e., when being uploaded/downloaded by a solution user, and at rest, i.e., while being stored on *CS* or on the ledger. The confidentiality of the *system metadata* could be violated if the content of the ledger is revealed to an unauthorized user. The confidentiality of the *cryptographic keys* can be violated if the user's smart card and credentials are compromised.

In our protocol, the confidentiality of the healthcare data is ensured first, by employing the *CS* that is assumed to be honest-by-curious. Therefore, we assume that *CS* will provide an access to the data only with respect to the corresponding permissions. Second, the confidentiality of the healthcare data is ensured by the security properties of the symmetric encryption algorithm applied to encrypt the data before uploading the data to the cloud server or the ledger. Therefore, neither non-authorized users (**ADV1-2**) nor the cloud server (**ADV3**) can decrypt the healthcare data.

To ensure the confidentiality of the system metadata, the system has to guarantee that caregiver *C* can learn only about the permissions specified for *C* (and nothing about the permissions specified by *P* for any other users). In our system, the chaincode implementation guarantees that *C* can only query the permissions that corresponds to him.

If the confidentiality of $SK^{symm}{}_{P}$ is violated, that of the encrypted data can also be violated. The confidentiality of the patient's key $SK^{symm}{}_{P}$ at rest is ensured by the pin code of the smart card or the phone known only to its owner. In transit, the confidentiality of the secret key is provided by the properties of the asymmetric encryption scheme: $SK^{symm}{}_{P}$ is encrypted with the public key of caregiver *C* ($PK^{E}{}_{C}$), with whom *P* wants to share his key.

Before any permission is defined by the patient and the secret key, $SK^{symm}{}_{P}$, is shared, the data can only be accessed (and deleted if needed) by the patient who uploaded the data. If the symmetric key of the patient is compromised (but $SK^{S}{}_{P}$ is still controlled by *P*), *P* can

temporary modify the access-control policy such that no data are shared by adding new permissions **Step 3**. The patient will need to generate another key, $SK^{symm'}{}_P$, to update the permissions and to share off-chain the fresh key (**Step 2**).

The **unlinkability** property in our system is defined as an impossibility for any adversary to link the system metadata and the corresponding patient's identity. Only the users authorized by the patient could link the patient's identity and his record stored on the blockchain. Unlinkability can be violated only if the confidentiality of the key $SK^{symm}{}_P$ is violated, as $SK^{symm}{}_P$ is used as a part of the input of a secure hash function used for pseudonym generation. Thus, the unlinkability property required in our system relies on the confidentiality of the $SK^{symm}{}_P$ and the following cryptographic properties of a secure hash function: pre-image resistance, strong collision-resistance, target collision-resistance, pseudo-randomness, and non-malleability. The patient's pseudonym can be generated only by the caregivers with whom the patient has shared his key $SK^{symm}{}_P$, as $SK^{symm}{}_P$ is used as a part of the input for pseudonym generation.

Validating nodes cannot link the access-control policy to the identity of a real person, as the *VN*s do not have an access to the patient's secret key $SK^{symm}{}_P$. Unless validating nodes collude with the caregivers (and in the system design it is assumed that the collusion is not possible between the caregivers and the validating nodes), linking the identity with the record stored on the blockchain is impossible.

3.6 Limitations

It is challenging to apply a relatively new technology, which is not yet framed by the laws and regulations, in a highly regulated healthcare. Even though the potential benefits of applying blockchain for healthcare are justified in Section 3.3 and a prototype of the proposed solution has been implemented, some technical limitations of our solution proposed in the chapter, as well as those related to the application domain can be highlighted below.

In our system model, we rely on a standard single trusted certification-authority. To avoid a risk of having a single point of failure, to relax the assumptions, to provide stronger security and to distribute the trust, alternatives to a single *CA* could be considered. An example of a scalable solution is presented in [STV+16]. The properties of the group signatures [BMW03] and anonymous credentials [BCKL08] could also be explored to address this limitation in future work.

The blockchain implementation (Hyperledger Fabric v0.6) employed in the prototype uses PBFT consensus protocol that provides excellent scalability in terms of the number of users; but this protocol has not been well explored in terms of the number of nodes (verified only up to few tens of nodes) [Vuk15]. A new version of Hyperledger Fabric (v1.0) is now adopted also in the prototype implementation. This version introduces several changes in the architecture (presented in Section 2.4.2), including an entity named orderer that is responsible for ordering the transactions in the blocks. In Hyperledger Fabric v1.0, PBFT consensus protocol has not

yet been employed for ordering the transactions. Therefore, the available architecture with only one orderer might re-introduce the risk of having a single point of failure of the system.

In the healthcare domain, emergency situations occur regularly, and the healthcare data might be required urgently. In case of an unavailability of the key, used to encrypt the patient's healthcare data, according to our approach it is impossible to access the data stored on the *CS*. An example of this situation is the following. An access-control policy can be defined in such a way so that only the patient possesses the encryption keys and that no caregiver from a medical institution (where the patient was delivered in an emergency situation) has a right to access any data about the patient. In this case, according to the design of our framework, if the patient is unconscious, it is impossible to update the permissions and grant access to the data to a caregiver. Robust and secure "break-glass" mechanisms for emergency situations are required to address this limitation.

According to the new General Data Protection Regulation in Europe, the patient has "the right to be forgotten". This right might be not compatible with the immutability principles of the current implementations of the blockchain technology used in this work: the patient cannot delete his record from the ledger. Applying different cryptographic techniques such as asymmetric encryption, proxy re-encryption, and defining correspondingly robust procedures for the keys management could be used to address such limitations and require further investigation.

Finally, the need for interoperability with the current infrastructures of the medical institution, as well as the challenge of providing usable interfaces for the patients and the medical doctors, could become an obstacle for the successful integration of such a system in the real-world settings. Further evaluation and testing of the different prototypes could be used to address these limitations and to guarantee the adoption of such a system.

4 Privacy-Preserving Utility-Aware Data Aggregation

In this chapter, we address the challenge of building an anonymized medical database from multiple sources. Our solution provides a possibility for data aggregation in a heterogeneous network of many clinical institutions and preserves the utility of the data and privacy of the patients. We present an algorithm for aggregating the healthcare data from multiple sources for research purposes and a protocol for improving the utility of the anonymized data in a distributed environment with database growth.

4.1 Introduction

While an anonymized database is being built from multiple sources of individuals' sensitive data, a person's privacy could be violated [PXW+07]. Even if the data are locally anonymized, their aggregation can still reveal sensitive information, especially if the data about an individual are distributed between different local databases [BLLW11, BS14]. Several models in the area of distributed privacy-preserving data publishing have already been proposed (i.e., pseudonymization [EII+10, XC14], secure multi-party computation (SMPC) [CJ05], microaggregation [SMBMS13], cloning [BLLW11]). However, these models significantly affect the utility of the data, therefore, an efficient independent release of the data from multiple sources and their aggregation without violation of privacy remains an open problem [GL13, PXW+07].

This problem is of a great interest, especially in the case of secondary use of the medical data. This includes the analysis of the patient's healthcare data in order for the enhancement of their healthcare experiences and the expansion of knowledge about different diseases and their appropriate treatments. Datasets that are built from multiple sources and contain health-related information about individuals are increasingly becoming "open". In this chapter, we answer the following question: How do we share and aggregate medical data for the research purposes while preserving privacy and utility of the data in a distributed environment?

Collecting medical data raises privacy concerns as these data are of a personal nature to the patient. In a medical setting, the following requirements have to be considered: the ability to

update the data about a patient (without creating multiple entries corresponding to the same person) and the possibility to recontact the patient through the caregiver who uploaded the data.

Our contribution of this chapter is the following. We present an algorithms for medical-data sharing and aggregation that enables healthcare professionals to release patients' data for research purposes while insuring patients' privacy. To achieve this we employ anonymization and pseudonymization techniques. We use generalization and suppression methods based on the taxonomies in the form of binary trees for data anonymization. We use symmetric encryption for generating pseudonyms. We then propose a protocol for improving the utility of the data and for preserving the required privacy level, we also formalize de-generalization criteria in distributed environment.

The rest of this chapter is organized as follows. In Section 4.2 we present the system model for aggregating the healthcare data in a privacy-preserving way. In Section 4.3, we focus on data aggregation: we describe our solution that employs pseudonymization and anonymization techniques for constructing a research database. In Section 4.4, we present de-generalization methods for improving utility of the anonymized data. Finally, we evaluate the utility (in Section 4.5), analyze the privacy and security of the proposed solution (in Section 4.6), and discuss the limitation of our system (in Section 4.7).

4.2 System Model

The system model (cf. Figure 4.1) is composed of (iii) the cloud server (CS) that hosting research databases, (iii) multiple caregivers ($C_1, \ldots C_N$) who provide the patient's data according to the patient's consent, and (iii) the certification authority, CA.

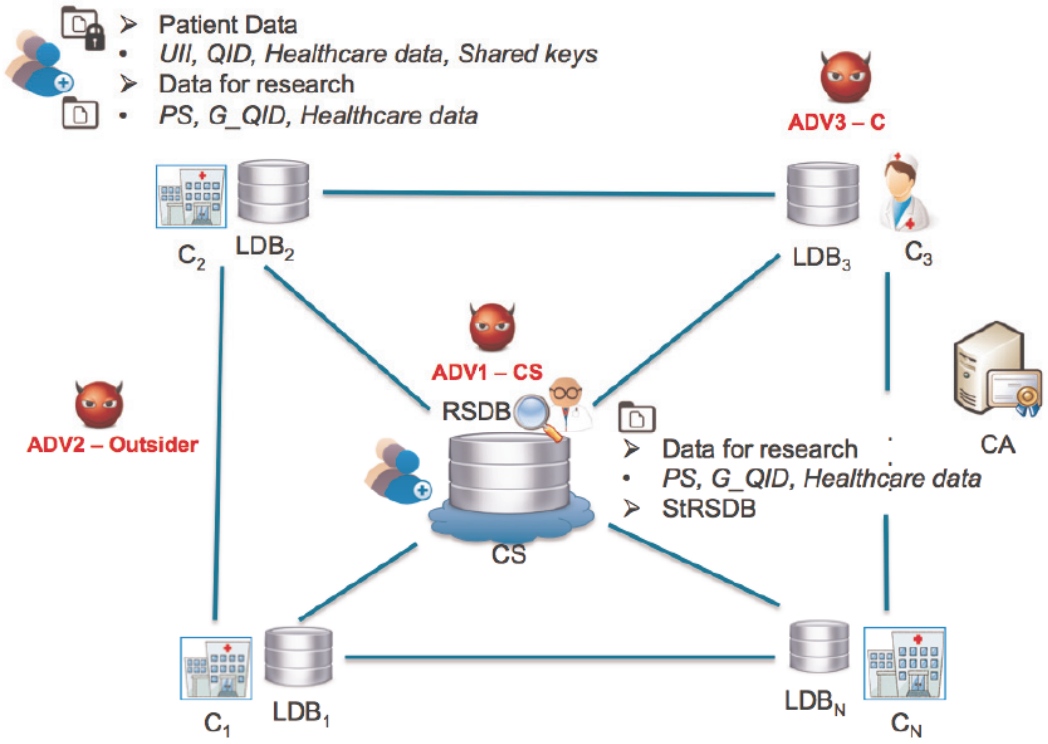

Figure 4.1 – System Model

A *Cloud Server, CS*, is a platform for the researchers and research institutions and is used to maintain research databases with patient's healthcare data while preserving patients' privacy. Multiple research databases can be hosted on the *CS*. For simplicity, we focus on creating just one $k-$*anonymous* research database, *RSDB*. The k-anonymity model has been extensively studied and used in practice. The background on $k-$*anonymity* and the review of existing approaches, together with their limitations, can be found in Section 2.3 of this thesis. *RSDB* contains the following data about a patient: a pseudonym (*PS*), anonymized quasi-identifiers (**G_QID**), and a subset of the healthcare data. The *CS* also stores information on the current state of *RSDB* (*StRSDB*), in particular, the number of records from each data source. We describe the data stored in *RSDB* and *StRSDB* in more detail in Section 4.3.2.

Caregiver C_i can be an independent medical practitioner, a medical institution or a hospital that maintains a local database, LDB_i, where the patient's data are stored. We assume that healthcare data that belong to the same patient could be stored in multiple local databases. The caregivers want to run a protocol to construct $k-$*anonymous* databases for research purposes. A symmetric cryptographic key $\mathcal{K}^S_{P,C_i}$ is shared between the patient P and every caregiver C_i, that P visits. These keys are stored on the patient's smart card or his phone and in the secured infrastructure of the medical institution where healthcare data are also stored. We assume that the patients' keys are securely shared with the caregivers (similar to the

approach discussed in the previous chapter), according to the access-control policy defined by the patient.

As in the previous section, we also assume here that for every patient, caregiver and the cloud server a standard *certification authority, CA* can generate certificates for a key pair for signing $(\mathsf{SK}^{\mathsf{S}}, \mathsf{PK}^{\mathsf{S}})$ and an encryption key pair $(\mathsf{SK}^{\mathsf{E}}, \mathsf{PK}^{\mathsf{E}})$. The secret keys can then be stored on a smart card, or a mobile phone with a chip.

Threat Model and System Assumptions

ADV1. We assume that the *CS* can be an honest-but-curious adversary. The *CS* will try to break the $k-$*anonymity* property of the *RSDB* in order to infer some additional information about the patients. We assume that the *CS* will not delete any data provided by the caregivers (the caregivers store the information about the data they uploaded to the *CS*, hence by removing the data, the *CS* will reveal his malicious behavior). We also assume that the data stored on the *CS* are backed up in case if the *CS* craches.

An outsider (**ADV2**), who is modeled as a passive adversary, can monitor the communications and updates of the *RSDB* to infer some information about the patients beyond the limit set up by the $k-$*anonymity* property of *RSDB*.

ADV3. We assume that a caregiver C_i is a Byzantine adversary. The local database, LDB_i, of C_i might not always be available, but due to the medical settings, C_i will not provide a false information about the patient. We also assume that the patient can revoke from a caregiver C_i the right to update the *RSDB* with the data about P. In this case however, the caregiver C_i will still be able to monitor the updates of the *RSDB* and will later be able to link the data that are uploaded by caregiver C_j, if the patient has allowed C_i to merge the pseudonym generated by C_i with the pseudonym generated by C_j, or vice versa.

As in Section 3.3 we use a single trusted certification authority. If it is compromised, then an adversary can create malicious users that could impersonate caregivers and could gain access to the private and symmetric keys shared between the patient and the caregivers. To relax the assumptions, to provide stronger security, and to distribute the trust, the single CA could be substituted by collective authority servers. An example of a scalable solution is presented in [STV+16].

We assume that secure communication channels have been established typically over HTTPS between all parties in the system; and that no collusion is possible between caregivers, a certification authority, and a cloud server; every message from a caregiver and the cloud server is digitally signed. We further assume that data from the network layers (e.g., IP address of the medical institution, of independent medical doctor) cannot be used to infer any complementary information about the patients. This assumption is reasonable as many users only access the Internet through a NAT gateway offered by their Internet provider, and could be relaxed if, for instance, the caregivers employ VPN services or anonymous networks (e.g., Tor) to access

the Internet.

Design goals
We define the following functionalities of our system: *i* Constructing $k-anonymous$ databases and update them with genuine healthcare data from distributed sources that might store healthcare data about the same patients.
(ii) Recontacting the patient through the caregiver that uploaded the data.
(iii) Avoiding multiple entries in the *RSDB* that correspond to the same patient when the patient's access control policy allows.
(iv) Improving the utility of the anonymized data with the database growth. While providing the functionality defined above, the system ensures **patient's privacy**. In particular, the following privacy guarantees (**PG**) should be provided:
PG1: Only anonymized healthcare data can be uploaded to the research database.Given that that anonymization is performed based on the current state of the research database, the system has to guarantee that the *CS* (**ADV1**) and an outsider (**ADV2**) learns nothing more about the patient during the anonymization process.
PG2: The unlinkability between the different pseudonyms of a same patient and between the patient's pseudonym and his data for any adversary is guaranteed. Linking and merging the pseudonyms can only be possible with respect to the access-control policy. Patient P can permit caregiver C_i to link the data already uploaded by C_j with the data generated by C_i. This, however, will mean that C_j will also be able to track the data he uploaded with the data uploaded by C_i.
PG3: the $k-anonymity$ of the *RSDB* should always be preserved, *(i)* at every step of constructing the database, *(ii)* populating it with the new records, and *(iii)* updating the records to improve the data utility.
The system has to ensure the following security properties:
Data integrity: every entity is ensured that the data stored in *RSDB* were not altered in transit or at rest by an adversary.
Authenticity: every entity is ensured that the data was sent by the claimed sender (C_i or CS).
Availability: the anonymized data are available on the *CS* from any where any time for the researchers.
Confidentiality: the disclosure of information (healthcare data, cryptographic keys) to an unauthorized individual is not possible.

4.3 Constructing Databases for Research Purposes

In this section, we first describe the data structure, then we present the algorithm for updating the *RSDB* with patents healthcare data that have been anonymized and pseudonymized for research purposes.

4.3.1 Solution Overview

We use the anonymization and pseudonymization techniques to create a $k-$*anonymous RSDB* and to update it from the distributed sources (multiple *LDB*s) in a privacy preserving way. Before uploading any data to the *CS*, a caregiver needs to verify the patient's consent: whether the data have to be anonymized or not. Then, any existing $k-$*anonymization* algorithms can be used to create an initial instance of *RSDB* with the data from one *LDB*. Then, for the updates in the distributed settings, we designed and implemented an algorithm that enables up to preserve the $k-$*anonymity* of the database and the privacy of the patient, whose information the *RSDB* is being updated with.

With every caregiver C_i that patient P visits, P generates a shared key, $\mathcal{K}^S_{P,C_i}$. The key is used to create the pseudonym for the patient. If the patient permits caregiver C_i to upload the data to the *CS*, the patient can decide whether he wants the data uploaded by C_i to be linked with the data that are already uploaded to the *CS* by another caregiver C_j. To enable C_i to link P's data with the data uploaded by C_j, P has to share the $\mathcal{K}^S_{P,C_j}$ with C_i. Sharing $\mathcal{K}^S_{P,C_j}$ with C_i will not enable C_j to link the pseudonyms before C_i links the pseudonyms at least once. However, once C_i has updated the *RSDB* by linking the pseudonyms generated by C_i and C_j, each of these two caregivers will be able to track the updates of the other.

For caregiver C_i, the algorithm for update the *RSDB* with a patient's record consists of the following main steps: *(i)* searching whether any information about the patient is already in the research database, *(ii)* verifying whether the pseudonyms could be merged, and *(iii)* generalizing the attributes of the record (quasi-identifiers) with respect to the outcomes of the previous steps and the current state of the *RSDB*. The current state of the *RSDB* is summarized and kept up-to-date in the separate data structure, *StRSDB*. In the next section, we present in detail the schema of the databases, notations (cf. Table 4.1) and data structure.

4.3.2 Notations and Data Structure

Hereafter, we describe the structure of the data that are stored in the databases.

Notations	Description
EHR	Electronic Health Record
CS	Cloud Server
CA	Certification Authority
LDB	Local Database
RSDB	Research Database
StRSDB	Metadata of the *RSDB*
P	Patient
UII	Uniquely identifiable information of the patient
C	Caregiver
$\mathsf{SK}^{\mathsf{S}}{}_{C_i}/\mathsf{PK}^{\mathsf{S}}{}_{C_i}$	Secret/public key of caregiver C_i for signing
$\mathcal{K}^{S}_{P_i,C_i}$	Key shared between patient P_i and C_j for pseudonyms generation and searching
PS^i	Patient's pseudonym, generated with the shared key $\mathcal{K}^{S}_{P_i,C_i}$
$r\,(PS)^t$	Row in the table t corresponding to the pseudonym PS of the patient
$\{qid_1,\ldots qid_Q\}$	Quasi-identifiers
QID	Values of the quasi-identifiers
G_QID	Generalized values of quasi-identifiers
Q	Number of quasi-identifiers, $\|\|\boldsymbol{QID}\|\|$
VGH_q	Value generalization hierarchy corresponding to qid_q, $q \in Q$
EQ_ID	Equivalence class ID
$N(PS)_i^{\boldsymbol{G\_QID}}$	Number of pseudonyms provided by caregiver C_i to the equivalence class with the ID constructed from the values ***G_QID***

Table 4.1 – Notations

UII	PS	QID		G_QID	
		age	gender	G_age	G_gender
John ...	PS^1	25	m	[25,38)	m
Mary ...	PS^2	39	f	[38,50)	f
Sam ...	PS^3	30	m	[25,38)	m
Max ...	PS^4	71	m	[50,75)	m
Emma ...	PS^5	45	f	[38,50)	f
Rick ...	PS^6	24	m	[25, 38)	*

Table 4.2 – Example of data representation in LDB_1 of caregiver C_1

Patient data consists of the following.

- *UII* – Uniquely identifiable information such as the combination of names, date of birth, social security number;

- ***QID*** – quasi-identifiers – a set of the values of the attributes ($\{qid_q\}$) that, when combined, can uniquely identify the person, for example, single-valued attributes are attributes such as age, gender, ZIP code; and set-valued attributes are attributes such as diagnosis codes;
- *Healthcare data* – healthcare information about the patient, used for primary care, including the data required for studies: for example, concentration measurements (time, measurement) – multiple attributes, that can be set-, or single-valued).
- *Shared cryptographic keys*, ($\mathcal{K}^S_{P,C_j}, j \in j \in [1..J], i \neq j$), – a set of the keys related to the patient and shared with C_i.

prevPS	PS	G_QID		EQ_ID
		G_age	G_gender	
	PS^8	[38,50)	f	{011,1}
	PS^{10}	[38,50)	f	{011,1}
PS^9	PS^{11}	[38,50)	f	{011,1}
	PS^{17}	[38,50)	f	{011,1}
	PS^4	[50,75)	m	{10,0}
	PS^7	[50,75)	m	{10,0}
	PS^{12}	[50,75)	m	{10,0}
	PS^{43}	[50,75)	m	{10,0}
	PS^{13}	[50,75)	m	{10,0}
	PS^{18}	[25,38)	*	{010,*}
PS^1	PS^9	[25,38)	*	{010,*}
	PS^6	[25, 38)	*	{010,*}

Table 4.3 – Example of data representation in 3-anonymous *RSDB* constructed from three sources.

Data for research purposes consist of the following:

- *Pseudonyms* – a set of artificial identifiers created to protect the identity of a patient when his data are released to the *RSDB*;
- ***G_QID*** = $\{G\_qid\}$ – a set of the generalized values of quasi-identifiers $\{qid_1, \ldots qid_Q\}$, constructed using corresponding value generalization hierarchy VGH_{qid_q}.
- **EQ_ID** – an equivalence class ID defined as a vector, where every coordinate represents G_qid in the form of a binary string with respect to the VGH_{qid_q}). An equivalence class contains records with the same set of ***G_QID***.
- A subset of healthcare data that will be used for research purposes and cannot be used to re-identify the patient to whom these data belong.

EQ_ID	N(PS)	$\{(C_i, N(PS)_{C_i})\}$
{011,1}	4	$(C_2,2);(C_3,2)$
{10,0}	5	$(C_1,1);(C_2,2);(C_3,2)$
{010,*}	3	$(C_1,1);(C_2,2)$

Table 4.4 – Example of data representation in *StRSDB* reflecting current state of 3-anonymous *RSDB* from Table 4.3.

StRSDB – is a table that characterizes the current state of the $k-$*anonymous RSDB*. For each equivalence class that is presented in the *RSDB*, *StRSDB* stores the following information: *N(PS)* – a number of entries with different pseudonyms from the *RSDB* associated with the same *QID* (number of elements in the equivalence class) and the sources of records (C_i – ID of the caregiver that uploaded the data, $N(PS)_i^{\boldsymbol{G\_QID}}$ – a number of records uploaded by caregiver C_i from the equivalence class with ID $\boldsymbol{G\_QID}$). Notice that as the *RSDB* is $k-$*anonymous*, $N(PS)_i^{\boldsymbol{G\_QID}} \geq k$ and $\sum_{i\in[1..N]} N(PS)^i_{\boldsymbol{G\_QID}} = N(PS)_{\boldsymbol{G\_QID}}$. Table 4.4 presents an example of *StRSDB*.

We also assume that each record is time-stamped and signed using the private key of the C_i that provided the data. Tables 4.2 –4.4 show the examples of the *LDB* (Table 4.2), *RSDB* (Table 4.3), and *StRSDB* – the metadata of *RSDB* (Table 4.4). As specified above, the healthcare data are stored in the *LDB* and *RSDB*. In the *LDB* – the data used in the context of primary care, including the data that are shared for the research purposes, and are sent to the *RSDB*.

Table 4.1 contains the notations we use in this chapter to present an algorithm for updating the *RSDB* from distributed sources while preserving its k-*anonymity*.

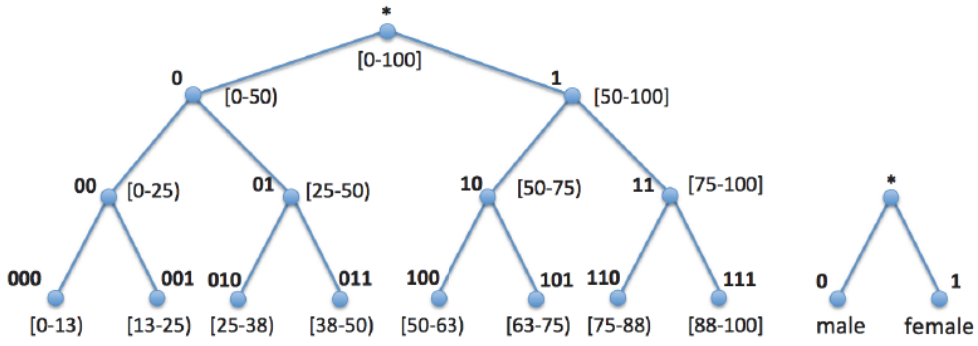

Figure 4.2 – Example of value generalization hierarchies for *qid*s: age and gender.

4.3.3 Algorithm for Updating *RSDB* from Distributed Sources

In this section, we present an algorithm that enables us to update the *RSDB* from multiple *LDB*s, while preserving the $k-$*anonymity* property of the *RSDB*. We ensure that if the patient permits caregivers to link the data originated from multiple *LDB*s, the corresponding caregivers

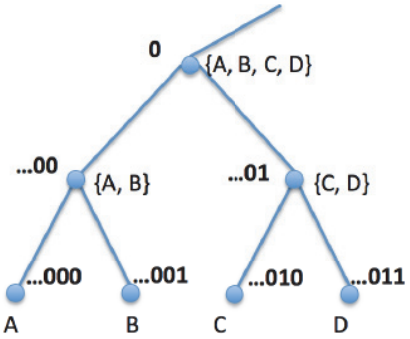

Figure 4.3 – Example of value generalization hierarchies for any set-valued *qid*.

will be able to update *RSDB* with the data about the patient, without creating multiple entries that correspond to the same person. Our solution also provides a possibility to recontact the patient through a caregiver who uploaded the data.

We assume that the *RSDB* is initialized with the locally anonymized $k-$*anonymous* database by employing generalization with hierarchical taxonomies defined for each *qid*. Several algorithms [GDLS14] could be used for constructing locally anonymized datasets for single valued attributes. An algorithm for achieving $(k, k^m)-$*anonymity* proposed in [PLGDS13] can be applied to the local dataset in case of the presence of set-valued attributes. Figure 4.2 presents an example of binary trees that are constructed for the single-valued attributes: age and gender. Figure 4.3 shows an example of representing a set-valued attribute.

Algorithm 2 Pseudonymization and anonymization of a patient's record

Input: $P, LDB_i, \{VGH_q\}, RSDB, StRSDB, \{\mathcal{K}^S_{P,C_j}\}, \mathbf{J}$
Output: $(PS^i, \boldsymbol{G\_QID}(PS^i))$

1: **if** C_i has $\{\mathcal{K}^S_{P,C_j}\}, j \in [1..\mathbf{J}]$ **then**
2: $\{\mathbf{t\_attr}(P)\} = \text{SEARCHPS}(\boldsymbol{UII}_P, RSDB, \{\mathcal{K}^S_{P,C_j}\}, C_i)$
3: **if** $\{\mathbf{t\_attr}(P)\} \neq \emptyset$ **then**
4: **if** $\| \{\mathbf{t\_attr}(P)\} \| = 1$ **then**
5: $(PS^P, \boldsymbol{G\_QID}(PS^P)) \leftarrow \mathbf{t\_attr}(P)$
6: **else**
7: $\{\mathbf{t\_attr}(P)\} \leftarrow \text{MERGEPS}(k, \{\mathbf{t\_attr}(P)\}, RSDB, StRSDB)$
8: $(PS^i, \boldsymbol{G\_QID}(PS^i)) \leftarrow$ the least generalized $(PS^u, \boldsymbol{G\_QID}(PS^u)) \in \{\mathbf{t\_attr}(P)\}$
9: **else**
10: $PS^i = \text{Enc.Encrypt}(\mathcal{K}^S_{P,C_i}, \boldsymbol{UII}_P)$,
11: $\boldsymbol{G\_QID}(PS^i) \leftarrow \text{GENER}(k, \{VGH_q\}, \boldsymbol{QID}, RSDB, StRSDB)$

Figure 4.4 – Pseudocode of the pseudonymization-and-anonymization algorithm

Figure 4.4 shows the pseudocode of the pseudonymization-and-anonymization algorithm applied to a record from the *LDB* before sending it as an update to the *RSDB*. We consider N Caregivers that can upload the data to *RSDB*. This algorithm is executed by a caregiver C_i every time C_i updates the *RSDB* with the data about patient P. We also assume that P has already shared a set of the keys $\{\mathcal{K}^S_{P,C_j}\}$, $j \in [1..\mathbf{J}]$ with C_i, where $\mathbf{J}$ is a list of caregivers P visited and the shared keys with whom P also shared with C_i. First, when performing an update, C_i has to check whether he can link the update with the P' pseudonyms and the quasi-identifiers that are already stored in the *RSDB*. Using these shared keys ($\{\mathcal{K}^S_{P,C_j}\}$, $j \in [1..\mathbf{J}]$), C_i has to check whether there are already some data that had been uploaded to *RSDB* by a caregiver C_j.

1: **function** SEARCHPS($\boldsymbol{UII}_P$, *RSDB*, $\{\mathcal{K}^S_{P,C_j}\}$, C_i, $\mathbf{J}$)
2: $\mathbf{D} \leftarrow$ SELECT (PS, $\boldsymbol{G\_QID}(PS)$) from *RSDB*
3: **for** all $j[1..\mathbf{J}]$ **do**
4: $PS^j = \leftarrow$Enc.Encrypt($\mathcal{K}^S_{P,C_j}$, $\boldsymbol{UII}_P$)
5: $\boldsymbol{G\_QID}(PS^j) \leftarrow$ SELECT $\boldsymbol{G\_QID}$ from *RSDB* WHERE (($RSDB.PS == PS^j, j \in [1..\mathbf{J}]$)$\vee$(($RSDB.PS == PS^d, d \in \mathbf{D}$)$\wedge$($\boldsymbol{G\_QID}(PS^d)$ is a prefix of $\boldsymbol{QID}$))) **return** $\{(PS^j, \boldsymbol{G\_QID}(PS^j))\}, j \neq i, j \in [1..\mathbf{J}]$

Figure 4.5 – Pseudocode of the SEARCHPS() function

1: **procedure** MERGEPS(k, {**t_attr**(P)}, *RSDB*, *StRSDB*)
2: **while** no pairwise merge is possible **do**
3: Choose $\boldsymbol{G\_QID}(PS^i)$ and $\boldsymbol{G\_QID}(PS^j)$ from {**t_attr**(P)}: $i \neq j$
4: **if** $N(PS)_{\boldsymbol{G\_QID}(PS^i)} - 1 \geq k$ **then**
5: **if** for all G_qid^i in $\boldsymbol{G\_QID}(PS^i)$: G_qid^i is a prefix of G_qid^j **then**
6: Update *RSDB*: $PS^i \leftarrow PS^j$, $PrevPS(PS^j) \leftarrow PS^i$, $\boldsymbol{G\_QID}(PS^i) \leftarrow \boldsymbol{G\_QID}(PS^j)$
7: Update *StRSDB*: $N(PS)_{\boldsymbol{G\_QID}(PS^i)} \leftarrow N(PS)_{\boldsymbol{G\_QID}(PS^i)} - 1$
8: {**t_attr**(P)} $\leftarrow$ {**t_attr**(P)} $\setminus$ (PS^i, $\boldsymbol{G\_QID}(PS^i)$)
9: **else**
10: **if** $N(PS)_{\boldsymbol{G\_QID}(PS^j)} - 1 \geq k$ **then**
11: **if** for all G_qid^j in $\boldsymbol{G\_QID}(PS^j)$: G_qid^j is a prefix of G_qid^i **then**
12: Update *RSDB*: $PS^j \leftarrow PS^i$, $PrevPS(PS^i) \leftarrow PS^j$, $\boldsymbol{G\_QID}(PS^j) \leftarrow \boldsymbol{G\_QID}(PS^i)$
13: Update *StRSDB*: $N(PS)_{\boldsymbol{G\_QID}(PS^j)} \leftarrow N(PS)_{\boldsymbol{G\_QID}(PS^j)} - 1$
14: {**t_attr**(P)} $\leftarrow$ {**t_attr**(P)} $\setminus$ (PS^j, $\boldsymbol{G\_QID}(PS^j)$)

Figure 4.6 – Pseudocode of the MERGEPS() procedure

C_i will either use, depending on the result of the function SEARCHPS(), the already existing pseudonym of P and corresponding generalized quasi-identifiers from the *RSDB*. Or C_i will create a new pseudonym for P using $\mathcal{K}^S_{P,C_i}$ and, taking into account the current state of *RSDB*, will apply the generalization function (pseudocode is shown on Figure 4.7) to create $\boldsymbol{G\_QID}$,

with which the healthcare data will be digitally signed by C_i and will be uploaded to the *RSDB*. If the SEARCHPS() function (the pseudocode is shown on the Figure 4.5) returns more then one tuple corresponding to P, i.e., multiple entries about P, C_i will apply the MERGEPS() procedure to check if there is a possibility to merge these records, without violating the $k-$*anonymity* property of *RSDB*. During the SEARCHPS() procedure, C_i checks if the data about the patient were uploaded by other caregivers. For this, C_i uses the keys that P shared with C_i: $\{\mathcal{K}^S_{P,C_j}\}$, $j \in [1..\mathbf{J}]$. C_i generates the pseudonyms by using the shared keys and queries the *RSDB* to obtain the list of pseudonyms and corresponding equivalence class IDs. It is important to ensure that only C_i learns this information. To guarantee this, C_i first queries the *RSDB* to obtain a list of the pseudonyms that are currently presented in the *RSDB*. Then he chooses among them a number of pseudonyms (depending on the value of anonymity parameter k and ***QID*** of the record) to use in dummy queries in order to not reveal to any adversary that the pseudonyms $\{PS^j\}$, $j \in [1..\mathbf{J}]$, belong to the same patient P.

1: **function** GENER($k, \{VGH_q\}$, ***QID***, *RSDB*, *StRSDB*)
2: Initialize $\boldsymbol{G\_QID} = \{G\_qid_q\}, q \in Q$ such that the value of every attribute (*G_qid*) is equal to the value of the root of VGH_q
3: $STOP =$ `false`
4: $VIEW \leftarrow StRSDB$
5: **for** every G_qid_q $q \in Q$ **do**
6: **while** ((G_qid_q is not a leaf) $\wedge$ ($STOP ==$ `false`)) **do**
7: $\boldsymbol{G\_QID}^0[q] = G\_qid_q + "0"$
8: $\boldsymbol{G\_QID}^1[q] = G\_qid_q + "1"$
9: Update *VIEW* as SELECT $\star$ from *VIEW* WHERE ((*StRSDB*.***G_QID*** is a prefix of $\boldsymbol{G\_QID}^0$) $\vee$ (*StRSDB*.***G_QID*** is a **prefix** of $\boldsymbol{G\_QID}^1$))
10: **if** $VIEW \neq \emptyset$ **then**
11: **if** SELECT $\star$ from *VIEW* WHERE ((*StRSDB*.***G_QID*** == $\boldsymbol{G\_QID}^0$) $\wedge$ (*StRSDB*.***G_QID*** == $\boldsymbol{G\_QID}^1$)) **then**
12: $\boldsymbol{G\_QID} \leftarrow \boldsymbol{G\_QID}^p$: $g(qid_q(PS)) = \boldsymbol{G\_QID}^p[q]$
13: **else**
14: $STOP =$ `true`
return ***G_QID***

Figure 4.7 – Pseudocode of the generalization function GENER()

The following could be an alternative to the dummy queries. In order to search for the pseudonyms corresponding to the same patient, a multi-key searchable encryption scheme, such as the solutions proposed in [BDCOP04, SWP00], could also be employed. An efficient model was proposed by Popa and Zeldovich [PZ13], however, it has been shown by Grubbs et al. that it does not imply security against either passive or active attacks [GMN+16]. Perillo et al. very recently proposed another solution to enable secure queries on encrypted multi-writer

tables [PPT17]. The possibility of employing this scheme for generating and searching over pseudonyms requires further investigations.

If the result of SEARCHPS() contains more than one pseudonym and ***QID***, C_i applies the MERGEPS() procedure. MERGEPS() enables us to check whether it is possible to merge pseudonyms that correspond to the same patients but are generated by different caregivers, according to the access control policy specified by the patient and the current state of *RSDB*.

The procedure takes as an input the parameter k, the set of pseudonyms, quasi-identifiers returned by SEARCHPS() procedure ($\{$**t_attr**$(P)\}$), and the databases: *RSDB* and *StRSDB*. To merge the data corresponding to two pseudonyms PS^1 and PS^2, with which the caregivers C_1 and C_2 uploaded the data of the same patient P, the following conditions have to be satisfied. First, every quasi-identifier from one of the sets ***G_QID***(PS^1), or ***G_QID***(PS^2), must be the prefix of the other corresponding quasi-identifier from ***G_QID***(PS^2), or ***G_QID***(PS^1), respectively. The second condition is to ensure that the equivalence class stays $k-$*anonymous* after merging the records. This update is equivalent to removing the record from one equivalent class and to decreasing the number of the entries in the *RSDB* corresponding to the same patient. However, note that we cannot guarantee that this procedure will result in having only one record corresponding to the patient, as the possibility to merge the pseudonyms depends on the access-control policy specified by the patient and on how the data originated from different sources (even if they are corresponding to the same patient) are generalized. The pseudocode of the procedure MERGEPS() is presented on Figure 4.6.

Caregiver C_i can perform a MERGEPS() procedure only before he makes the first update of the *RSDB*. Therefore, in order to be able to merge pseudonyms later on, the following strategy can be applied. According to the access-control policy specified by the patient, a caregiver who possesses the largest number of the keys man perform SEARCHPS() and MERGEPS() procedures every time after de-generalization function is executed. This will decrease the number of pseudonyms with which the information about the patient had been uploaded by different caregivers.

Figure 4.7 shows the pseudocode for GENER() function that is performed based on the current state of the *RSDB* by caregiver C_i to create the generalized attributes ***G_QID*** for the record of the patient, whose data have not yet been uploaded to the *RSDB*, or whose access control policy does not allow merges. For instance, the data about the patient P might have been uploaded by the caregiver C_x, but caregiver C_i who for the first time wants to upload the data about patient P does not possess the key $\mathcal{K}^S_{P,C_x}$. Apart from the anonymity parameter, k, the procedure takes as input quasi-identifiers, $\{$***qid***$\}$, generalization taxonomies for each quasi-identifier, $\{VGH_q\}$, and the databases *RSDB*, *StRSDB*, Examples of the taxonomies are shown in Figure 4.2 and 4.3. The function returns the least generalized equivalence class to place the record of P without disclosing corresponding to *PS* values of ***QID*** from *LDB*.

Each ***qid*** is being generalized one after another (the order is based on the importance of the ***qid***) using the top-down generalization approach [LDR06, HN09]. First, every ***qid*** is assigned

with the most generalized value (roots of the corresponding taxonomy trees). Then, we choose one qid_q and iterate over the nodes, checking if de-generalization is possible at every level of the taxonomy, until we reach a leaf, or the stopping criterion is true. The stopping criterion is defined as the impossibility to de-generalize the quasi-identifier at least one more step. This, in turn, depends on the current state of the *RSDB*, i.e., the equivalence classes, where the record could be placed.

For each step, we verify whether the *StRSDB* contains the equivalence class IDs with the $\boldsymbol{G\_QID^0}$ and $\boldsymbol{G\_QID^1}$, such that every G_qid_q is a prefix of the corresponding value qid_q of the record: $\boldsymbol{G\_QID^0}[q] = G\_qid_q + \text{"0"}$, $\boldsymbol{G\_QID^1}[q] = G\_qid_q + \text{"1"}$, where $G\_qid_q, q \in Q$ represents the current level of generalized quasi-identifier qid_q of the record. It is important to check both classes, as if we check only one $\boldsymbol{G\_QID^+}[q]$, corresponding to the prefix of the least generalized value of the attribute qid_q, and the query of *StRSDB* returns an empty set, the anonymity of the record will be violated. The rsult of this query immediately reveals a more fine-grained value of an attribute qid_q.

4.4 Improving Data Utility

First, we present an overview of our one-step de-generalization approach, which improves the utility of the records from an equivalence class with respect to one attribute. Next, we describe our proposed privacy-preserving protocol for one-step de-generalization. Finally, we extend our proposed approach and formalize the criteria for the de-generalization of a dataset. Our experimental evaluation follows.

4.4.1 One-Step De-generalization

The problem of minimizing data generalization while preserving privacy in the case of incremental updates was first studied and addressed in [PXW+07]. The authors propose incremental updates and the refinement of the anonymized database. However, this solution, first, does not take into account the possibility of existing overlapping populations in the databases (i.e., information about the same individual could be present in multiple data sources). Second, this approach could not be used in distributed settings, as the refinement requires an access to the non-anonymized initial version of the data, which is impossible in a distributed environment without privacy violation. The former issue is addressed by the psuedonymization approach and presented in Section 4.3.3. To address the latter, we formulate a de-generalization problem in a distributed setting and provide a protocol for distributed one-step de-generalization.

One-step distributed de-generalization can be defined as a process of constructing two equivalence classes, eq_0 and eq_1, from an existing k-anonymous equivalence class EQ such that *(i)* the generalization level of an attribute of eq_0 (eq_1) is decreased by one with respect to the generalization level of the same attribute in EQ, *(ii)* eq_0 and eq_1 are k-anonymous, and *(iii)* both could contain the records from multiple sources.

prevPS	PS	G_QID		EQ_ID
		G_age	G_gender	
	PS^{8}	[38,50)	f	{011,1}
	PS^{10}	[38,50)	f	{011,1}
PS^{9}	PS^{11}	[38,50)	f	{011,1}
	PS^{17}	[38,50)	f	{011,1}
	PS^{4}	[50,75)	m	{10,0}
	PS^{7}	[50,75)	m	{10,0}
	PS^{12}	[50,75)	m	{10,0}
	PS^{43}	[50,75)	m	{10,0}
	PS^{13}	[50,75)	m	{10,0}
	PS^{81}	[50,75)	m	{10,0}
	PS^{18}	[25,38)	*	{010,*}
PS^{1}	PS^{9}	[25,38)	*	{010,*}
	PS^{6}	[25, 38)	*	{010,*}

Table 4.5 – Example of the data representation in the *RSDB* constructed from multiple (four) sources after one single-record update (highlighted with different coulor).

EQ_ID	N(PS)	$\{(C_i, N(PS)_{C_i})\}$
{011,1}	4	$(C_2,2);(C_3,2)$
{10,0}	**6**	$(C_1,1);(C_2,2);(C_3,2);(C_4,1)$
{010,*}	3	$(C_1,1);(C_2,2)$

Table 4.6 – Example of *StRSDB* before one-step de-generalization, after a single-record update. The second equivalence class is now updated

EQ_ID	N(PS)	$\{(C_i, N(PS)_{C_i})\}$
{011,1}	4	$(C_2,2);(C_3,2)$
{101,0}	3	$(C_1,1);(C_2,1);(C_3,1)$
{100,0}	3	$(C_2,1);(C_3,1);(C_4,1)$
{010,*}	3	$(C_1,1);(C_2,2)$

Table 4.7 – Example of *StRSDB* after one-step de-generalization. The second equivalence class is now split in two.

For instance, according to the binary trees (shown on Figure 4.2), the records from the equivalence class with the attributes coded as {10;0} can be de-generalized by splitting them into groups with more fine-grained description of one of the attributes (cf. Table 4.6 and Table 4.7). We can consider (based on the current state of Table 4.5) forming the following groups [50-63) and [63-75) coded as "100" and as "101", respectively, with respect to the age attribute; and

the gender stays generalized the same way:

$$\{10;0\} \Rightarrow (\{010;0\} \wedge \{011;0\}) \vee (\{010;0\}) \vee (\{011;0\}) \tag{4.1}$$

However, the following constraints need to be taken into account:

$$\begin{cases} k^p \geq k, \forall p \in \{0,1\}, \\ \sum_{p=0}^{1} k^p = K, \\ \sum_{p=0}^{1} a_t^p = A_t. \end{cases} \tag{4.2}$$

Equation 4.2 formalizes the requirements from the definition above. First, the number of records in any of the groups (k^0, k^1) must be greater then the parameter k required for $k-$*anonymity*. Second, the sum of the number of records in both groups after splitting must be equal to the number of records in the equivalence class before splitting (K). This can be guaranteed only by taking into account the number of records that came from different sources $a_t, t \in [1..T]$, where $T-$ is a number of sources that contributed the records to the equivalence class.

The solution is straightforward if there exists t, such that $(a_t^0 \geq k) \wedge (a_t^1 \geq k) \wedge (a_t^0 + a_t^1 = K)$. In other words, if C_t is the only contributor equivalence class EQ, hence local de-generalization is possible. For the case when one-step de-generalization is not possible locally, we propose the following protocol for collaborative privacy-preserving one-step de-generalization.

4.4.2 Protocol for One-Step De-generalization

Our protocol consists of the following phases: initialization, (**INIT**), evaluation of the possibility to de-anonymize, (**EVAL**), and updating the databases (**UPD**), if de-anonymization is possible. We assume that cryptographic keys are generated during the setup phase of the public-key encryption, functional encryption, and the threshold encryption schemes. The *CA* is employed to generate and provide certificates of public keys.

During the **INIT** phase the *CS* chooses the parameters for one-step de-generalization: an equivalence class ID and an attribute. This choice also defines the participants: caregivers who contributed the records to this equivalence class. Then, the caregivers compute their shares of the information required for the *CS*, to evaluate in a privacy-preserving way, whether the de-generalization is possible.

- **Step 1**: The *CS* queries *StRSDB* and chooses an equivalence class ID (EQ) with the number of records K, such that $K > 2 \times k$, the name of the attribute, q, and its current generalized value G_qid_q.

- **Step 2**: The *CS* sends to every C_t, $t \in [1..T]$, the following data: an ID of the equivalence class EQ, the attribute qid_q, the list of the pseudonyms from the equivalence class EQ, and a share of the master private key $\boldsymbol{mpk}$={params, pk_t} to be used in the functional encryption scheme. The IDs of the contributors are known to the *CS*, as the healthcare data are signed by the caregivers when they upload the data.

- **Step 3**: Every caregiver C_t verifies that in his *LDB* there are the records with the corresponding pseudonyms. Then, C_t de-generalizes the current version of G_qid_q one step furhter, based on the non-generalized values qid_q of the records with the given pseudonyms and the taxonomy tree.

- **Step 4**: Caregivers $\{C_1, \dots C_T\}$ compute a shared randomness r to be used for the functional encryption scheme by using the Shamir secret-sharing scheme.

- **Step 5**: Every C_t, $t \in [1..T]$ computes the number of the pseudonyms from the given equivalence class EQ that can be de-generalized, such that $G\_qid_q \leftarrow G\_qid_q$ + "0" (a_t^0), and the number of the pseudonyms from EQ that can be de-generalized, such that $G\_qid_q \leftarrow G\_qid_q$ + "1" (a_t^1).

- **Step 5**: Every C_t, $t \in [1..T]$ computes the commitment of the shared randomness c_0^t, as well as the encrypted values of the number of records computed at the previous step: $c_t^0 \leftarrow \mathsf{IPEnc}^{\mathsf{E}}.\mathsf{Encrypt}(\boldsymbol{mpk}, a_t^0)$, and $c_t^1 \leftarrow \mathsf{IPEnc}^{\mathsf{E}}.\mathsf{Encrypt}(\boldsymbol{mpk}, a_t^1)$

- **Step 6**: Every C_t, $t \in [1..T]$ sends the commitment c_0^t, and the values encrypted during the previous step ($\{c_t^0, 0\}$, and $\{c_t^1, 1\}$) to the *RSDB*.

During the following step (**EVAL**), the *CS* evaluates whether the de-generalization of the records from EQ is possible, without accessing the numbers of the pseudonyms from EQ provided by different caregivers: a_t^0, and a_t^1, $t \in [1..T]$.

- **Step 7**: After receiving the information from T caregivers, the *CS* constructs a vector $\mathbf{c}^0 = \{c_0^t, c_1^0, \dots, c_T^0\}$ and a vector $\mathbf{c}^1 = \{c_0^t, c_1^1, \dots, c_T^1\}$ in order to compute the sum of the numbers of records from the different sources that can be de-generalized to the same ***G_QID***.

$$\begin{cases} k^0 \leftarrow \mathsf{IPEnc}^{\mathsf{E}}.\mathsf{Decrypt}(\boldsymbol{mpk}, \mathbf{c}^0, sk), \\ k^1 \leftarrow \mathsf{IPEnc}^{\mathsf{E}}.\mathsf{Decrypt}(\boldsymbol{mpk}, \mathbf{c}^1, sk), \\ (k^0 = K) \vee (k^1 = K) \vee \big((k^0 \geq k) \wedge (k^1 \geq k)\big), \\ \big(k^0 + k^1 = K\big) \vee (K - (k^0 + k^1) \geq k) \end{cases} \tag{4.3}$$

Section 2.2 contains a detailed description of the decryption step of the functional encryption scheme that is employed at this step: If it is possible to improve the utility of the data by de-generalizing the records ((3.3) is `true`), the *RSDB* and *StRSDB* have to be updated accordingly (**UPD**). However, the *CS* does not yet know how the records from EQ must be split between

two new equivalence classes eq_1 and eq_0, and the *CS* does not know which caregivers will contribute to one or both them. To learn how the equivalence class have to be split, *CS* sends a request to every C_t. If some of the sources fail to provide this information, the third condition of the equation (3.3) might not hold anymore, therefore, the privacy of the records from the other sources, and the anonymity of the *RSDB*, can be violated. To prevent this, a secret-sharing scheme is used.

- **Step 8**: The *CS* sends to the contributors of EQ the following information: the EQ, the list of the pseudonyms, the attribute q, and its current generalized value G_qid_q, requesting to provide the corresponding one-step de-generalized version of the value of the attribute, computed at **Step 3**.
- **Step 9**: Every $C_t, t \in [1..T]$ encrypts the updates $\{(EQ, PS, q, G\_qid_q^+)\}_t$ (where $G\_qid_q^+$ is a one-step generalized version of G_qid_q from *RSDB*) by using a threshold-encryption scheme: $upd_t \leftarrow \mathsf{TPKE.Encrypt}(PK_{CS}, \{(EQ, PS, q, G\_qid_q^+)\}_t)$. Every C_t also constructs the decryption share $\mu_t \leftarrow \mathsf{TPKE.ShareDecrypt}(PK, t, SK_t, upd_t)$, and sends (μ_t, upd_t) to the *CS*.
- **Step 10**: The *CS* verifies the validity of every share, $\mu_t, t \in [1..T]$: $\mathsf{TPKE.ShareVerify}(PK, VK, \mathscr{C}, \mu_t)$, and combines them to decrypt the updates $(\{upd_1, \ldots, upd_T\})$ from the caregivers $C_1, \ldots, C_T : \{(EQ, PS, q, G\_qid_q^+)\}_t \leftarrow \mathsf{TPKE.Combine}(PK, VK, upd_t, \{\mu_1, \ldots, \mu_T\}), t \in [1..T]$. Then, the *CS* updates the *RSDB* and the *StRSDB* accordingly.

4.4.3 Distributed De-generalization Problem

Striving for maximization of the utility of the data, we would like to de-generalize the data as much as possible. To formalize the de-generalization, we first need to define utility and its measure. Measuring the utility of the released data is a difficult task that also heavily depends on intended data usage of the dataset, as the utility evaluation focuses on the particular type of knowledge that is to be extracted [BLJ08]. Currently, no single utility measure is broadly accepted [BLJ08]. Existing utility measures used for microdata releases are reviewed in [HDFF+12]. Bertino et al. in [BLJ08] survey the utility measures used in privacy-preserving data mining.

We use the following notations to define the utility measure for our scenario. For each equivalence class, we define a vector $\mathbf{L}$. Every coordinate of this vector represents the level of de-generalization of the records in the equivalence class, with respect to an attribute q, $q \in Q$. This vector consists of the lengths of the binary strings – codes describing generalized version of the attribute value, normalized by the depths of the corresponding taxonomy trees (d_{VGH_q}): $L = \{l_{qid_q}\}$. The vector is constructed based on the taxonomies defined for every attribute, VGH_q. It is intuitive that the closer the generalized value to the leaf on VGH_q is, the more information is preserved. In some research settings, the importance of the attributes are different. Therefore, we, also introduce a weight, or an importance of the attribute: $\mathbf{w} = \{w_q\}$

for every attribute q. This parameter will be used to define the order of the attributes, when applying one-step de-generalization in our implementation.

In our scenario, we define utility as follows. The utility of the dataset is the average utility of all the equivalence classes from this dataset. The utility of the records from the same equivalence class is equal. Therefore, we define the utility of a record i as u_i, and as U_D-utility of a dataset as follows:

$$u_i = \frac{1}{N} \sum_{q=1}^{Q} w_q \times \frac{l_{qid_q}}{d_{VGH_q}}, \tag{4.4}$$

$$U_D = \frac{1}{D} \sum_{i=1}^{D} u_i. \tag{4.5}$$

Then, the goal of de-generalization is defined as follows: To substitute an equivalence class with ID ***G_QID*** with the set of equivalence classes, {***G_QID+***}, such that *(i)* utility of the dataset is maximized, *(ii)* all the equivalence classes are $k-$***anonymous***, and *(iii)* no records are lost during de-generalization.

Equation 4.6 formalize the requirements and constraints described above in the following way:

$$\begin{cases} U_D \text{ is max}, \\ k_{G\_QID+} \geq k, \forall \textbf{G\_QID+} \in \{\textbf{G\_QID+}\}, \\ \sum_{\forall \textbf{G\_QID+} \in \textbf{G\_QID+}} k_{\{G\_QID+\}} = k_{G\_QID}. \end{cases} \tag{4.6}$$

The different equivalence classes in the *RSDB* are independent. Therefore, to satisfy Equation 4.6, we apply to each equivalence class from the *RSDB* our one-step distributed de-generalization solution, subsequently for every quasi-identifier.

4.5 Evaluation

In this section we present an evaluation of the approach presented in this chapter: constructing $k-$***anonymized*** database from multiple sources (Section 4.3) and improving data utility by using a de-generalization approach (Section 4.4). The purpose of the evaluation of the utility of the resulting *RSDB*. For the security and privacy evaluation, we rely on the theoretical analysis presented in the following section.

For the evaluation, we used reference dataset "Census" that contains 1080 records with numerical attributes [BDFMS02]. However, as we use taxonomies, our method is also applicable to the categorical attributes. This test dataset was used in the European project CASC [BDFMS02],

and in [DFGN10, SCDFSM14, DDFS02]. Like in [DFGN10, SCDFSM14], we used the following attributes: FICA (social security retirement payroll deduction), FEDTAX (federal income tax liability), INTVAL (amount of interest income), and POTHVAL (total other persons income).

We compare our solution with the following baseline approaches: centralized settings and a distributed baseline approach. The centralized approach is equivalent to applying local anonymization once to the whole dataset. In this case, the whole non-generalized dataset is available when anonymization is applied, thereby preserving the utility. However, with this approach the amount of the data is limited (as there is only one data source), and to collect the same amount of data, it would take more time than than if multiple sources were available.

The distributed baseline approach consists of the following three steps: *(i)* the dataset is split among multiple sources, *(ii)* an anonymization algorithm is applied locally, and, *(iii)* the anonymized datasets are merged to construct *RSDB*. Notice that this baseline approach does not take into account the challenge of preserving patient's privacy if the data about the same patient are presented in multiple sources.

In the solution proposed in this chapter we assume that an initial *RSDB* needs to be constructed first. We conducted experiments with initial datasets of two sizes: 50, and 200 records. Then, the rest of the dataset was split evenly among 2, 4, and 8 data sources, which is equivalent to having 3, 5, and 9 data sources, correspondingly, for the baseline approach (with one source always having 50 or 200 records). An important consideration is that in our approach, any number of sources can be used (upper-bounded by the difference between the total number of records and the size of the initial dataset), as every update consists of only one record.

The implementation of the anonymization and the de-generalization methods were done in the Java programming language and the experiments were conducted on a PC with 2.8 GHz Intel Core i7 processor and a 16 GB main memory.

Figure 4.8 shows the utility of the databases built from the initial databases of two different sizes by using the following approaches: *(i)* the distributed baseline approach (independent local anonymization of multiple databases), *(ii)* the centralized baseline approach (anonymization applied to the database of 1080 records), and *(iii)* our solution (data anonymization that improves the utility with the database growth). We evaluate two ways of choosing the attributes for our de-generalization approach: *(i)* similar to the baseline approach (always following predefined order) and *(ii)* by choosing randomly the order of attributes for applying one-step de-generalizaton for every equivalence class. We show the averaged results for the different number of records in the initial *RSDB*, and the different values of the anonymity parameter k. To compute the utility of the resulting datasets, we use the utility measure introduced in Section 4.4.3.

With the increasing number of the sources with the baseline approach, the size of the local database decreases. Hence, the utility decreases as well: the less data are available, the more generalization occurs. A straight-forward countermeasure is to not release the data before the

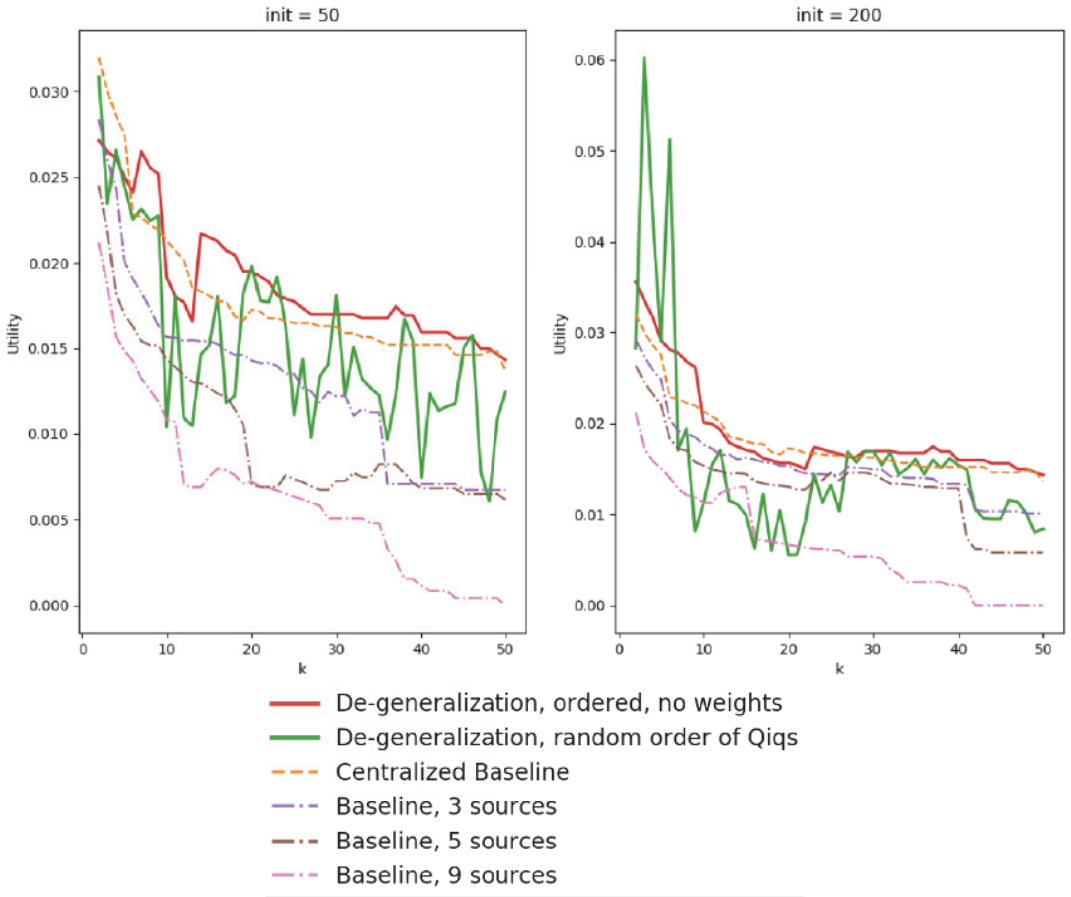

Figure 4.8 – Comparison of the utility of the data while using approach, centralized, and distributed baselines.

Different number of sources (2 sources, 4 sources, 8 sources); the weight of an attribute $w_q = 0.25$, different number (init) of records in the initial database, and different parameter k.

size of the local databases reaches a certain number of records. However, this can significantly slow down the process of data aggregation that is a crucial prerequisite before beginning the research. With our solution, the records can be released from any number of sources, one by one in a privacy-preserving way.

In addition, our approach provides privacy guarantees when the information about the same patient is presented in multiple databases. The risk of violating the privacy of the patient can arise with the baseline approach, depending on the background information available about the patient. In our approach we address this issue by applying pseudonymization (Section 4.3.3).

If we relax the requirements and assume that the information about a patient can be present only in one dataset, using the baseline approach is "safe". Then, we can experimentally compare the privacy levels of the resulting datasets constructed using our approach, the

centralized method and the baseline approach. The privacy level of the database is defined as the size of the smallest equivalence class in the database. Usually, the desired level is chosen upfront and has to be kept in the resulting database, as it enables us to bound the probability of the de-identification of a person. Due to the nature of the dataset, multiple quasi-identifiers, and taxonomies, constructed based on the requirements of the research study, when applying anonymization, it is not always possible to construct the equivalence classes with exactly k records.

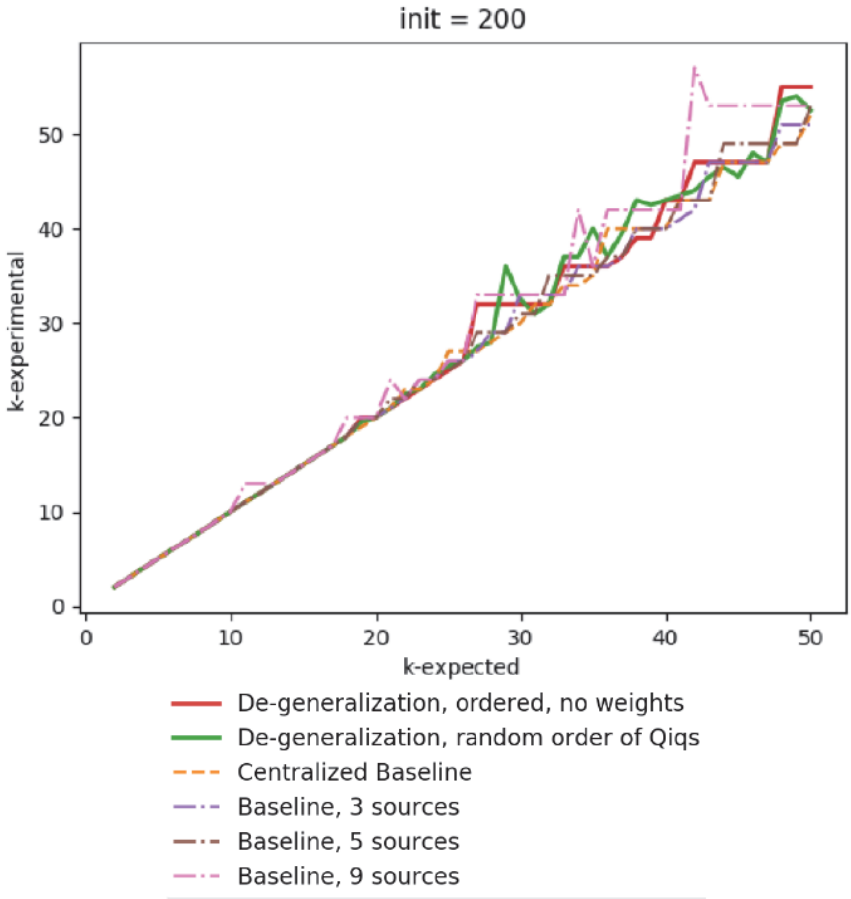

Figure 4.9 – Actual privacy level vs. required privacy level
Comparing actual privacy level when applying our approach with centralized approach and distributed approach (2 sources, 4 sources, 8 sources) with the weight of an attribute $w_q = 0.25$, size of the initial database (init): 200 records, and different parameter k.

Figure 4.9 shows the privacy level of the database constructed during experiments versus the required privacy level defined as parameter k, for different values of k. As shown in the figure, for all the values of k, the experimental values are never smaller than required. This shows that the $k-$*anonymous* property is preserved when applying our solution. The best-case scenario, in terms of utility requirements (the trade-off between privacy and utility), is when the experimental and required $k-$*anonymity* levels are the same. We can expect that if the

utility of the baseline approach decreases compared to our solution, the privacy level of the resulting database will be higher than in the case of our approach. However, Figure 4.9 depicts that the privacy level of the baseline approach does not differ significantly from the privacy level of the database constructed using our approach.

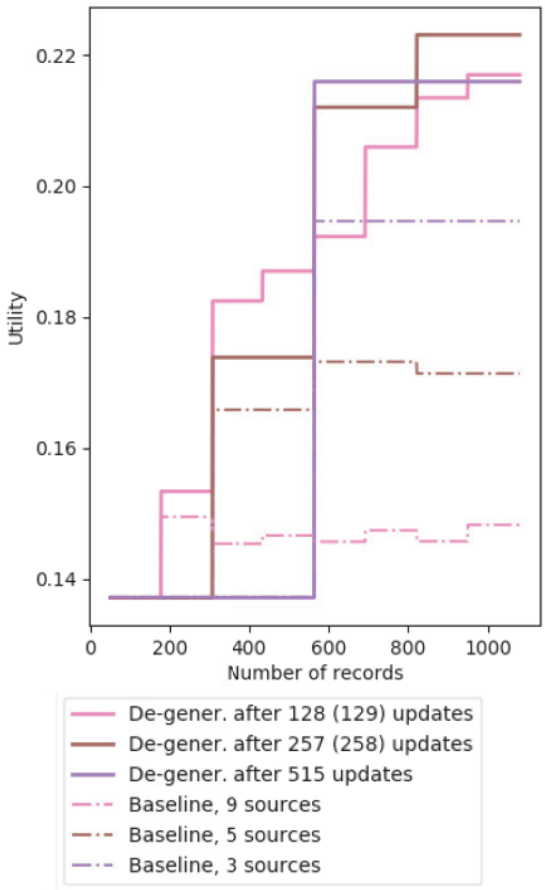

Figure 4.10 – Utility of the database when performing de-generalization after different number of updates vs. merging locally anonymized databases. For (de-)generalization the order of attributes was kept, no weights used.

Figures 4.10–4.11 show how the utility of the dataset changes with the growing number of the records in the *RSDB*. We apply de-generalization not only after the database has been updated with all the records, but we also try to perform de-generalization step after updating the database with different number of records. For this scenario, we selected the parameters as follows: k=5, the size of the initial *RSDB* = 50 records. Then, we split equally the rest of the dataset into multiple parts (chunks). We run the experiments for different chunk sizes, which is equivalent of updating the initial *RSDB* from multiple data sources (2, 4, and 8) by using the baseline approach.

The choice of the chunk size also influences the utility, as Figure 4.11 shows. After updating the *RSDB* with a single record, it might not be possible to de-generalize the dataset because

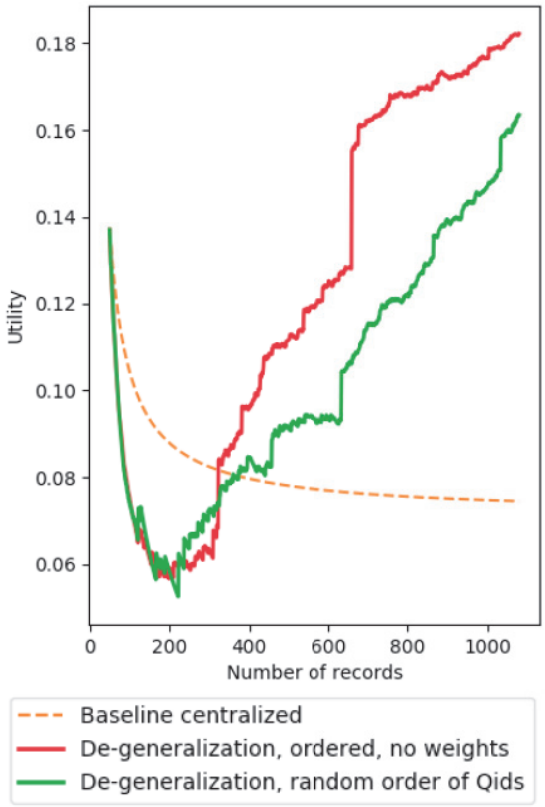

Figure 4.11 – Utility of the database when performing de-generalization after single-record update vs. updating one locally anonymized database without refining the data (centralized baseline). For (de-)generalization, the order of attributes was kept, no weights were used.

a certain amount of the records has to be accumulated in the equivalence class to be split. Hence, this approach cannot provide the best utility of the resulting dataset.

Other advantages of our approach are demonstrated in Figure 4.12 and Figure 4.13. Figure 4.12 shows how the utility changes with respect to the attribute weight, or importance (w_q). De-generalization is performed by applying one-step de-generalization after every single update (if possible) taking into account the weight of the attribute. Figure 4.12 also shows the utility of the dataset for different specifications of the utility vector, **w**. This demonstrates that the utility can be adjusted, depending on the nature of the attributes and the corresponding taxonomy trees.

Figure 4.13 demonstrates that the proportion of the records with more than two suppressed attributes decreases faster with the database growth when we apply our approach, compared to the centralized one.

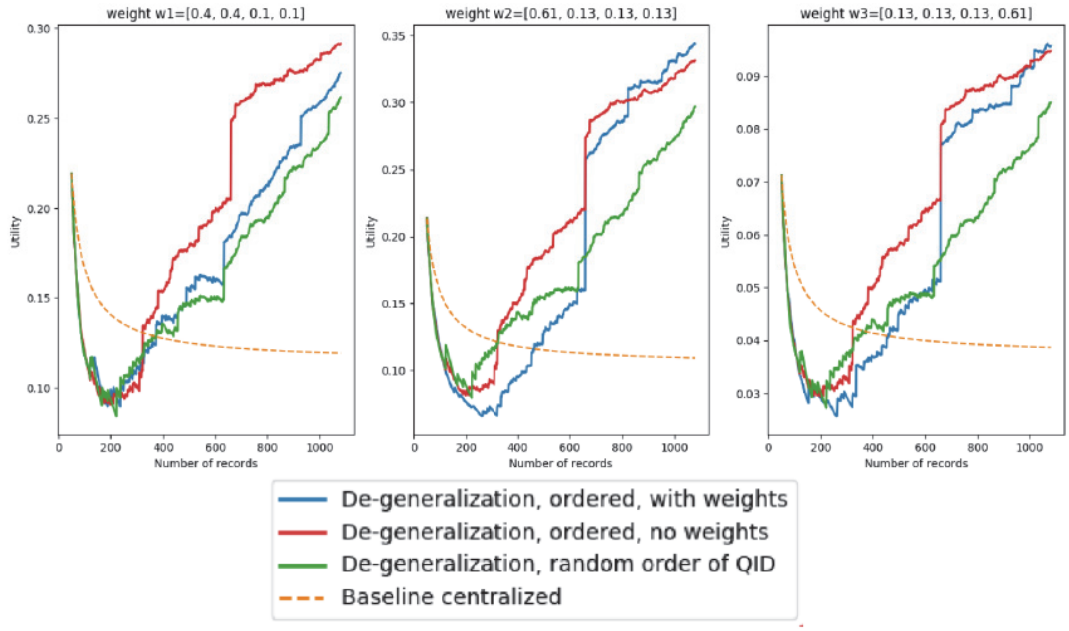

Figure 4.12 – The effect of taking into account pre-defined importance of the attributes, w_i, on the utility of the dataset, during dataset construction with k=5, size of initial $RSDB = 50$ records.

4.6 Privacy and Security Analysis

In this section, we analyze the privacy and security of our solution for constructing the $k-anonymous$ database from multiple sources and improving its utility with the database growth. We recall the privacy and security goals specified in Section 4.2 and we analyze them.

To ensure the **patient's privacy**, we guarantee that the process of the anonymization of the initial *RSDB* and of the the subsequent single-record updates does not leak any information about the patient, apart from the resulting generalized attributes.

Our approach provides the following privacy guarantees (defined in Section 4.2) that the patient's privacy is preserved during all the steps of the algorithm for constructing the *RSDB* and the protocol for improving its utility:

- No data about the patient beyond the $k-anonymized$ data are leaked (**PG1**);
- The unlinkability between multiple pseudonyms corresponding to the same patient, and the unlinkability between the patient pseudonym and his identity for any anauthorized users is ensured (**PG2**);
- The *RSDB* preserves its $k-anonymity$ at any step of the proposed solution (**PG3**).

The patient's privacy could be violated during the following steps: when the first initial version

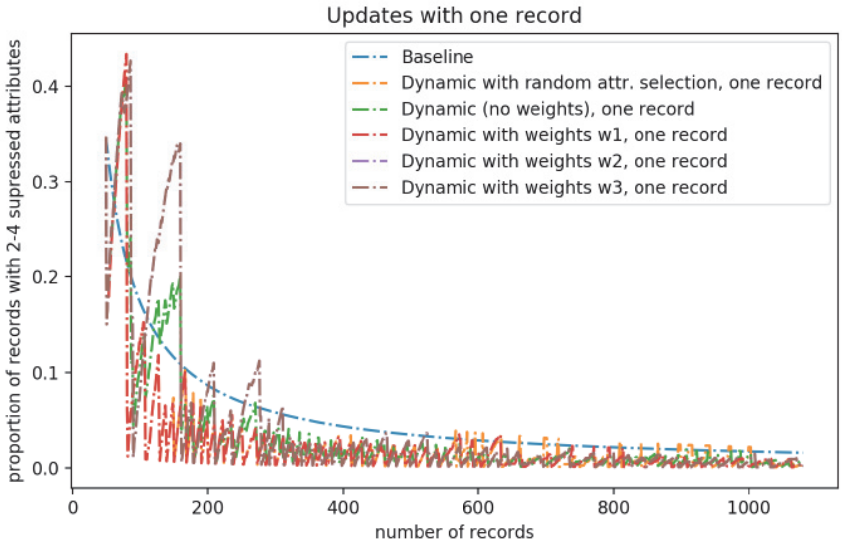

Figure 4.13 – Proportion of records with 2-4 suppressed attributes for different size of the dataset k=5, size of initial *RSDB* = 50 records.

of the database is released, when the database is updated with a single record, and when the data from *RSDB* are being de-generalized. Below we describe how we achieve the privacy guarantees at every step.

For the *release of the initial RSDB*, we rely on the correctness of the anonymization algorithm that is applied locally on an LDB_i . In our solution we use top-down Mondrian multidimensional recoding model [LDR06] with the generalization trees. The anonymity of the resulting database **PG3** is guaranteed based on the correctness of such algorithm. Furthermore, if the anonymization is performed locally, the *CS* (**ADV1**), the outsider (**ADV2**), or any other caregiver (**ADV3**) can learn nothing more than the initial version of the $k-$*anonymous RSDB*. The data in the *RSDB* are pseudonymized, and pseudonyms are generated using symmetric encryption. The initial *RSDB* is built locally, hence it contains only one pseudonym per patient. In this case, to provide **PG2**, we need to ensure that only the authorized users can link the identity of the patient with his pseudonym. Using the security properties of symmetric encryption we can guarantee **PG2**, as long as the confidentiality of the secret key (discussed further in the chapter) used for pseudonym generation is not violated. According to the assumptions (Section 4.2), we ensure that apart from the quasi-identifiers and pseudonyms, only a subset of the healthcare data (such as drug concentration measurements) that cannot be used to de-identify the person can be sent to the *RSDB*, thus (**PG1**) is guaranteed.

When performing*a single-record update*, a caregiver first has to search whether he can update the patient's record in the *RSDB* but without creating another record corresponding to the same patient (SEARCHPS() function is presented in Section 4.3.3). When searching over the *RSDB* for the pseudonyms corresponding to the same patient, C_i uses dummy queries in order to not to reveal to any adversary that the set of the pseudonyms $\{PS^j\}$, $j \int \in [1..\mathbf{J}]$, belongs to

the same patient P. This enables us to prevent unlinkability between multiple pseudonyms corresponding to the same patient for non-authorized users, hence achieving **PG2**.

The unlinkability between a patient's pseudonyms and his identity is provided by using shared keys for pseudonyms generation; the keys are shared between the patient and the caregivers. The pseudonyms are generated by applying symmetric encryption to the uniquely identifiable information of the patient. Therefore, similarly to the case of constructing the initial *RSDB*, using the security properties of symmetric encryption and confidentiality of the cryptographic keys, we can guarantee that only authorized caregivers can link the pseudonyms to the record corresponding to the patient.

When updating *RSDB* with a single record corresponding to the patient, we need to guarantee that the privacy of this patient is preserved (particularly, **PG1**).To do so, we employ a top-down generalization approach for every attribute by using a binary tree (generalization taxonomy). At every step of the GENER() function (Section 4.3.3) a caregiver queries the *StRSDB* to check whether the record can be assigned to one of the two equivalence classes that are one-level more de-generalized with respect to an attribute. If we only check one equivalence class, where we are intended to place the record, $\boldsymbol{G\_QID}^{+}[q]$, corresponding to the prefix of the least generalized value of the attribute qid_q, and the query returns an empty set, the anonymity of the record would be violated, and a more fine-grained value of an attribute q (qid_q) would be leaked. To prevent revealing the more fine-grained value of the attribute of the record, the presence of both classes is verified.

There are two possible outcomes of a single-record update: (iii) the information is merged with the pseudonym that is already in the database and (ii) the new record is added in one of the equivalence classes. We assume that the initial *RSDB* is $k-$*anonymous*, and this implies that every equivalence class has at least k records. Increasing the number of records in an equivalence class can only increase the level of privacy and, therefore the $k-$*anonymity* of the *RSDB* after updates is preserved and **PG3** is attained.

Finally, we provide a privacy guarantee (only **PG3** is relevant) for the last step, *improving the utility of the data with the database growth*.

The $k-$*anonymity* property could be at risk at the evaluation steps of the de-generalization protocol (Section 4.4.2). To eliminate the risk, we use the functional encryption scheme to securely evaluate whether a split of the equivalence class is possible without violating the $k-$*anonymity* of the *RSDB*. The security of the functional encryption scheme is based on the semantic security of the underlying public-key encryption scheme under randomness reuse. To share a random number among the contributors to the same equivalence class, we use the Shamir secret-sharing scheme that guarantees that only authorized coalitions can recover the secret, whereas any other sets of participants cannot obtain any posterior information about the possible value of the secret.

By using the functional encryption scheme with randomness reuse, we guarantee, that neither

C_i, *CS* or an outsider can access the information about how many records will be provided from a contributor to one of the two newly constructed equivalence classes. The *CS* is able to evaluate only whether it is possible to construct new equivalence classes, based on the results of the summations of all the inputs from the contributors.

When the actual split of the equivalence class occurs, **ADV3** (a caregiver) can be off-line. At this step, $k-$*anonymity* is based on the correctness and security of the threshold encryption scheme. Employing the threshold encryption scheme enables us to ensure the following decryption policy: the *CS* can decrypt only if all the contributors have sent their updates. This guarantees that the newly constructed equivalence classes are $k-$*anonymous* based on the evaluation step and on the assumption that the caregivers do not provide false information and do not collude.

Data integrity and **authenticity** are guaranteed when every entity is ensured that the data stored in the *RSDB* were not altered in transit or at rest, by an adversary and that the data was sent by the claimed sender (C_i or *CS*).

We assume that the healthcare data stored in the *LDB* cannot be altered by any adversary, as the data are stored within the secured infrastructure of the medical institution. However, the integrity and authenticity of the anonymized healthcare data could be violated when the data are stored in the *RSDB*, or when they are in transit from the *LDB* to the *RSDB*. The integrity and authenticity of cryptographic keys could be violated when the secret keys are stored and used by all the entities in the system, and when the shared key between the patient and caregiver C_i, ($\mathcal{K}^S_{P,C_i}$) is shared with another caregiver C_j.

The data integrity and authenticity are guaranteed by the correctness and unforgeability of the digital-signature algorithm used when uploading the data to the *RSDB*. They also rely on the integrity, authenticity, and confidentiality (discussed further in the chapter) of the cryptographic keys. A digital signature is also used during the steps of the de-generalization protocol to guarantee the integrity and authenticity of the messages exchanged between the caregivers and the *CS*.

The integrity and authenticity of the private keys used for digital-signature generation and the shared keys between the patient and caregivers rely on the security of the pin code of the smart card or the phone (where the keys are stored). The authenticity and integrity of the encryption key for pseudonym generation ($\mathcal{K}^S_{P,C_i}$), when shared by the patient with another caregiver C_j, rely on the properties of the digital signature scheme, as well as on the integrity, authenticity, and confidentiality of the cryptographic keys of the users (SK^S_U,PK^S_U).

The **availability** of the anonymized data is guaranteed for the researchers when the data are available on the *CS* from any where at any time. We underline that our system provides a possibility to update the *RSDB* with just one record without violating the privacy of the patients. Therefore, without delays, new data can become available in an anonymized form and increase utility for the researchers.

The data might not be available if the *CS* is down, or if the *CS* maliciously erases the data provided by the caregivers.

As mentioned in Section 4.2, the availability of the data is guaranteed by the backup services. According to the threat model, we assume that the *CS*, being modeled as an honest-but-curious adversary, will not delete the data provided by the caregivers (caregivers store the information about the data they uploaded to *CS*, thus by removing the data, *CS* will reveal his malicious behavior).

The rapid availability of the new data is ensured by the design of the algorithm for updating the *RSDB*. The research dataset can be updated with just one record from any caregiver and in a privacy-preserving way. There is no need to delay the data release for research purposes till a certain number of records is collected to perform anonymization locally.

Confidentiality ensures that the disclosure of information (healthcare data, cryptographic keys) to an unauthorized individual is not possible. The confidentiality of the healthcare data can potentially be violated if a caregiver sends non-anonymized data to the *RSDB* or if the confidentiality of the keys is violated and a pseudonym is decrypted by a non-authorized user.

The confidentiality of the uniquely identifiable information used to generate the pseudonyms is guaranteed by applying a symmetric encryption scheme. The confidentiality of the shared key between the patient and caregiver C_i, $\mathcal{K}^S_{P,C_i}$, relies on the pin code of the smart card or the phone. If a patient loses his smart card or his phone, the keys can be recovered from the corresponding caregivers that treat the patient.

The confidentiality of the key shared with C_i, $\mathcal{K}^S_{P,C_i}$, when sharing it with another caregiver C_j, is guaranteed by the security property of the public-key encryption scheme used to encrypt the key before sending it to C_j.

4.7 Limitations

Our solution has the following limitations; they can be addressed in future work.

When a process of distributed-data aggregation starts, the initial $k-$*anonymous RSDB* is required to be constructed from one of the *LDB*s. The utility of this database also depends on the number of records in the *LDB*.

The de-generalization taxonomy trees used for (de-)generalization of the quasi-identifiers have to be defined before anonymization and cannot be modified after the initial $k-$*anonymous RSDB* is built.

As mentioned in Section 4.3.1, we assume that if a patient authorizes one caregiver to merge the patient's data with the data about the same patient and provided by another caregiver, the latter would be granted the same permission with respect to the data provided by the

former. Even if the possibility to update *RSDB* was revoked, both will continue to be able to see that the patient's data are updated. Even though *RSDB* is updated with the anonymized data, only the fact that that update was or was not performed can potentially create a risk for the patient's privacy. To avoid the risk during the procedure of merging the pseudonyms, the previous pseudonym can be encrypted together with the information about the caregiver that created this pseudonym. The ciphertext and a parameter that indicates how many times the pseudonym had been updated will be stored in the *RSDB*. Then, it will be possible to find the caregiver who initially uploaded the data (e.g., in case of legal issues), through the caregiver(s) that merged the pseudonyms.

The generalization step (the GENER() function of the algorithm presented in 4.3.3) requires going one by one through all the *qid*s. However, we assume that the number of quasi-identifiers in the *RSDB* is not high and that they are ordered based on their importance (defined by the vector **w**) with respect to the requirements to the *RSDB*.

We also assume that the data from the network layers (e.g., IP address of the medical institution, or independent medical doctor) cannot be used to infer any complementary information about the patients. Even though we already assume that an anonymous communication service such as Tor [DMS04] can be used, we argue that the identity of the caregiver could still potentially be used by the *CS* to infer some information such as the location of the caregiver. Then, we can assume that there is a correlation between the inferred location of the caregiver and the addresses of his patients. Consequently, if the patient's address is one of the quasi-identifiers in *RSDB*, and its generalized value is high-level, this correlation can be employed to probabilistically derive a more fine-grained value of the patient's address. This can lead to the violation of the anonymity property of *RSDB*, hence the privacy of one or more patients.

As it was mentioned in Section 4.3.1, we assume that using a smart card or a mobile phone to store cryptographic keys is not be a burden for the vast majority of users. However, we cannot exclude the risk that the secret key can be compromised due to the human factor. One example is as follows. If the key $\mathcal{K}^{S}_{P,C_i}$ of P is compromised, then an adversary can break the pseudonym of P and infer the information about P. If the number of records in the equivalence class that contains a record with the compromised pseudonym is exactly k, the $k-$*anonymity* of *RSDB* will be violated by an adversary who possesses a compromised key. In this case, it is also impossible to simply remove the record, because the *RSDB* will not be $k-$*anonymous* anymore. Subsequently, a new key has to be generated, and the pseudonym will be updated. However, the new information about patient P cannot be merged with the data about P that is already in the *RSDB*. In the case of merging, an active adversary who monitors every update will be able to link the compromised pseudonym and the new one, hence will be able to monitor the updates in the future.

Another limitation, also related to the human factor, is that we cannot guarantee that a caregiver always provides genuine medical data to the *RSDB*. We cannot verify that there is no intentional or non-intentional alteration of the healthcare data of the patients. We only

assume that caregivers do not have any intentions to sabotage the research process or to provide false healthcare information for constructing the *RSDB*.

Interoperability in eHealth Part II

5 Data Exchange for Precision Medicine

Therapeutic drug monitoring (TDM) is a key concept in precision medicine. The goal of TDM is to avoid therapeutic failure or toxic effects of a drug due to insufficient or excessive concentration related to the variability between patients. In this chapter, we focus on the integration of *TUCUXI* – an intelligent system for TDM. *TUCUXI* was developed to be used in medical practice, to assist clinicians in taking dosage-adjustment decisions in order to optimize drug concentration levels. This software is currently being tested in CHUV (Lausanne University Hospital in Switzerland). We describe the software and present how we achieved its integration in clinical workflow. The modular architecture of the software allows us to plug in a module enabling data aggregation for research purposes. This is an important feature in order to develop new mathematical models for drugs and to advance TDM. We also discuss ethical issues related to the use of an automated decision support system in clinical practice, in particular, if it allows data aggregation for research purposes.

5.1 Introduction

Millions of people have to take a variety of medications every day. Unfortunately, treatments are not always effective in all patients. One reason for this is that different patients absorb, metabolize and eliminate drugs differently. Patients' response to a drug may depend on genetic makeup, age, body size, presence of kidney or liver diseases, drug-drug interactions, time of the day, etc. Therefore, the drug dose may be either insufficient, and the patient will not benefit from the treatment, or excessive, which may cause serious toxicity. This is especially relevant to critical medications such as anti-cancer or anti-HIV drugs that both have a narrow therapeutic range and a poorly predictable relationship between the dosage prescribed and the drug concentration in the patients' blood.

For example, for the antiretroviral drug *Rilpivirine* used in the treatment of HIV patients, the target minimum concentration is 44ng/ml. However, it has been shown that on average four patients out of ten have Rilpivirine concentrations below the target. Therefore, a risk of insufficient efficacy exists in 40% of the patients treated with the standard dosage [ABG+17].

The problem of inappropriate dosing has been reported in medical literature [ABG+17, CABC+12] and the TDM approach has been proposed as a corrective measure. TDM involves the measurement of the drug concentrations in biological samples and the adjustment of the drug dosage in order to improve drug efficacy and to reduce related toxicities [LBC16, GWM+12, MW14]. TDM has evolved to become an important tool that is used for administration of antiarrhythmic and psychiatric drugs, anticonvulsants, anticancer agents, immunosuppressants, and antifungals [KL09].

Pharmacokinetic and pharmacodynamic (PKPD) models for numerous drugs are being developed by clinical pharmacologists to describe how the body handles a drug in terms of absorption, distribution, metabolism, and elimination (PK models). PD models, in turn, describe how the drug affects the body by linking the drug concentration profile to one or several efficacy metrics. PKPD models are ideally suited to summarize the background knowledge necessary to adjust the drug dosage in a given patient [KL09].

However, in everyday clinical practice, it is fairly difficult for a clinician to make use of these PKPD models available only in scientific literature and to apply them in every specific patient's case. A consultation with a pharmacologist may not be arranged shortly and will require the pharmacologist to collect a sufficient amount of information from the patient's clinical history before issuing a valid recommendation for dosage adjustment. Still, an appropriate correction of the drug dosage can be of critical importance if the concentration exposure is significantly away from the targets ensuring optimal treatment efficacy and tolerability.

In order to address these issues, an automation of TDM is proposed. Existing software tools for TDM have been surveyed by Fuchs et al. in [FCT+13]. However, an intelligent system integrated in everyday clinical practice, allowing precise and rapid evaluation and adjustment of drug dosages, and simultaneously making the data available for the development of new PKPD models, is still missing.

Our development had to address the following challenges:

- *Interdisciplinary collaboration.* Mathematical models developed by researchers in clinical pharmacology need to be embedded in a user-friendly software suitable to be used by medical doctors/pharmacologists.
- *Ergonomy.* The software has to efficiently help medical doctors and pharmacologists by being well suited to the actual processing flow of TDM requests faced daily.
- *Medical device certification.* According to current regulation, such a system is a medical device and as such needs to be certified in order to ensure proper functionalities without risks of harming patients.
- *Interoperability.* The software requires seamless insertion into the existing network of electronic medical records, laboratory information system, and other medical applications, thus raising issues related to different interfaces, data formats, complexity of

clinical dataflow etc.

- *Data aggregation for research.* The collection of population data from the daily use of the software is ideally suited to improve the existing models and to develop new models for drug candidates for TDM. However, patients' data are sensitive, studies have very different scopes, and data aggregation is time consuming.

- *Ethical issues.* Automatisation of the data processing raises a question of medical liability regarding highly sensitive aspects of patient-data management such as dosage decisions.

In this chapter we present *TUCUXI*– a software that was developed and integrated into the clinical data flow in order to provide automated dosage evaluation and adjustment decisions to assist pharmacologists and medical doctors. The modular architecture of the software enables us to plug in dedicated modules for data aggregation for research purposes. These data are to be used by researchers to improve or develop PKPD models for TDM.

TUCUXI is an interactive tool designed to guide a user through the process of TDM. In particular, *TUCUXI* allows to calculate population, a priory and a posteriori percentiles, unlike the most advanced existing solutions such as*DoseMe* and *NextDose*. Moreover, the authors in [AMM17] claim that presently available software (such as *DoseMe*, *insightrx* and *NextDose*) is still sufficiently complex and requires training to enable rapid use at the bedside by healthcare professionals. In contrast, *TUCUXI* has a very intuitive user interface that makes the software easy to use by non-pharmacologists, by general practitioners, and, possibly, by educated patients. It can be integrated in clinical practice, and can also be connected to the research database.

The advantages of our solution are the following. First, it provides ***personalized dosage adjustment advice*** based on reference population PKPD data, patient's individual characteristics and concentrations previously observed, if available. Second, it ***optimizes TDM procedures*** by enabling medical professionals to process large numbers of requests; therefore it contributes to extend the use of TDM and the number of patients that may benefit from TDM services. Third, it provides an interface with other clinical applications, and could be easily ***integrated into primary care and also used for medical research.*** Finally, our solution also helps to ***decrease the risk*** of human mistakes.

The rest of the chapter is organized as follows. In Section 5.2, we present current non-automatized processes of routine TDM in hospital settings and its difficulties. We describe the developed software in details in Section 5.3 along with its integration into the clinical data flow in Section 5.4. In Section 5.5, we discuss how the software enables data aggregation for research purposes. In Section 5.6, we discuss ethical issues related to the use of an automated software in clinical practice, to its validation, and to the aggregation of the sensitive medical data in order to improve TDM and therapeutic outcomes.

5.2 TDM in Clinical Practice

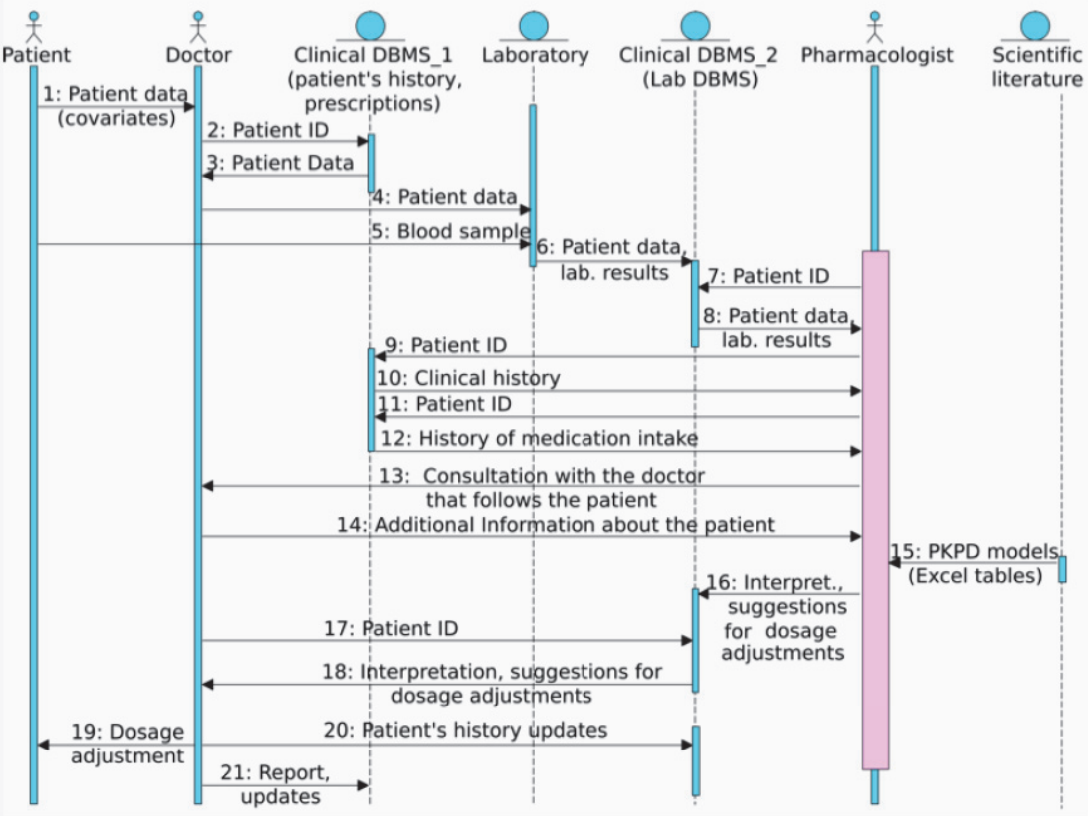

Figure 5.1 – TDM in clinical practice: non-automated process

Non-automated therapeutic drug monitoring, as currently practiced in most places, is a comprehensive and rather slow process. We studied the related procedures in actual medical practice at a University Hospital, where clinical pharmacologists work on a daily basis to provide dosage adjustment advice to medical doctors working either in the same hospital or elsewhere in the country.

The sequence diagram shown on Figure 5.2 (a) describes non – automated TDM[1]. The process begins when a patient starts receiving treatment with a drug considered to require TDM. In this case, after a couple of intakes, a blood sample is drawn to monitor the plasma concentration of the drug. The medical doctor in charge of the patient prescribes the test and requests to send the patient's sample together with appropriate data to the laboratory, where concentration measurements are performed (steps 1 – 5 of the diagram on Figure 5.2a). Then, next step (6): the data are transferred into a laboratory information system, where they are stored and will be accessed by the pharmacologist in charge of clinical interpretation of the measurement

[1]In order to simplify graphical representation of the processes on both sequence diagrams on the Figure 5.2 we do not show existing interfaces and proxies that can be modeled as boundary object and control objects and placed between actors and entity objects on the UML diagram.

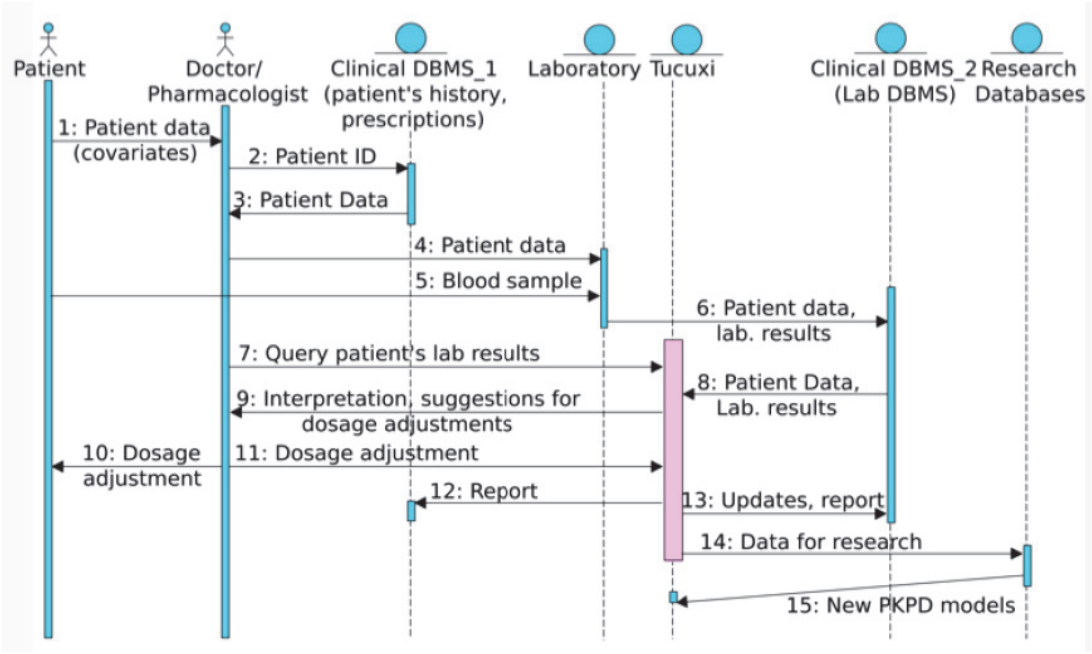

Figure 5.2 – TDM in clinical practice: *TUCUXI* (TDM software) integrated in clinical practice.

result, after its validation.

When the pharmacologist receives a request to interpret a drug concentration value, as already mentioned above, he needs to collect a certain amount of medical information regarding the patient's case. To do so, he may need to access multiple databases that store information about the the patient's clinical history, medication records, laboratory results, etc. (steps 7 – 12). The pharmacologist will also need to refer to PKPD models for the drug existing in the scientific literature (as step 15 indicates it). If any information about the patient is missing, a phone contact with the medical doctor that initiated the TDM request is also needed (steps 13 – 14).

When all the information is collected, the pharmacologist elaborates an interpretation of the laboratory result (16). At present, this is essentially made on an empirical basis. Some rare pharmacologists rely on models built up with custom tools (such as Excel), which can accommodate patient's medical characteristics and just one or sometimes two concentration measurements. Only a small minority of centers regularly use one of the commercially available dedicated software tools, considered of insufficient ergonomy for implementation in everyday routine [FCT+13]. The results and updates are sent to the hospital electronic medical record, which the physician that initiated the TDM request can now access (as presented by steps 16 – 21).

While working on each particular patient case, a pharmacologist has to access multiple sources of data, perform various estimations or calculations, switch contexts. Not only are current

procedures slow and poorly efficient, but they also may increase the risk of human mistakes, which could significantly affect patient's condition.

When clinical pharmacologists are not available in the hospital, most laboratory results are sent without interpretation to the physicians that have requested TDM. Most physicians tend to translate TDM results into dosage adjustments according to an empirical trial-and-error strategy – which should not be entirely denigrated though, as it often fits clinical needs to a sufficient extent. In exceptional cases another institution will be contacted (such as the Hospital we are collaborating with). However, this may cause even longer delays due to the need to transfer the data, to clarify missing details about patient's history (due to the lack of interoperability between different hospitals, e.g., data stored in an hospital can only be accessed internally due to their sensitive nature), and to elaborate the pharmacological interpretation requested.

In order to optimize highlighted steps 7 – 20 of the diagram on Figure 5.2a, and to tackle the difficulties listed above, we propose to integrate a TDM software in the clinical data flow. The software provides an interface that guides the user through the TDM process, provides required information about the patient and employs PKPD models built up using population data and integrated into the software. This allows not only to interpret the current concentration value but also to predict future drug exposure, and to suggest a personalized dosage adjustment for the patient. Figure 5.2b presents the sequence diagram of the data flow using TDM software *TUCUXI* integrated in clinical practice. One can notice that the flow is simplified (steps 7 – 13), and also includes populating a research database for building up new PKPD models (steps 14 – 15).

5.3 *TUCUXI* and Embedded Mathematical Models

A software helping the clinicians in their daily practice can be conceived as a standalone software embedding a graphical user interface (GUI) or as a service hidden behind a client. We present here the GUI version, and we will discuss the service possibilities in Section 5.6.

5.3.1 Software Description

TUCUXI core capabilities can be segmented into three main parts:

1. Computation of concentration percentiles and comparison with therapeutic targets, based on the patient's dosage regimen and clinical characteristics, as well as on reference PKPD data ;
2. Computation of concentration predictions based on the same data confronted with the patient's observations ;
3. Suggestion of dosage adjustments in order to drive the resulting concentration exposure

into the therapeutic targets.

The general architecture is made of layers. The first layer is pure mathematics, implemented in a very optimized way. It ensures the three core capabilities listed above. The second layer (processing layer) is responsible for translating Domain Model Objects into mathematical objects and then calling the math functions. On top of this second layer the GUI exploits the processing layer and an interconnection module to create the final software.

The following section gives some insight corresponding to the three main mathematical software parts.

Concentration calculation.

Prediction of drug concentration can be done through model-free systems (e.g., Support Vector Machine [SBY+14]) based on population data, or through model-based systems. The model-free approach may be well suited for predicting concentration at a specific time (typically time of measure), but often suffers from limits of applicability (depending on the method, concentrations below zero could be calculated). The model-based approach is more suited for continuous curves, and is used by a majority of pharmacologists (also because it allows to *explain* the calculation in a physiological way).

The concept is to model the human body in terms of compartments exchanging substances. The drug concentration is typically measured in blood, so a central compartment corresponds to the entire blood veins and arteries. Models can have more than one compartment, and if such, the muscles are a typical second compartment. Taking a drug is seen as an input to a specific compartment, and the drug elimination corresponds to a flow leaving one of the compartments. Figure 5.3 represents a 2-compartment model for bolus injection.

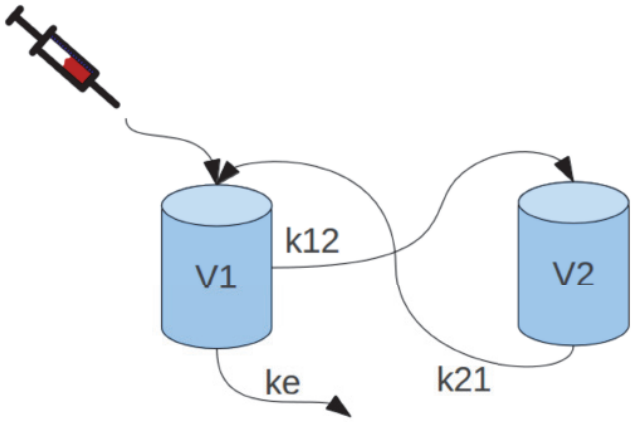

Figure 5.3 – 2-compartment model

Constants ke, $k12$, $k21$ are used to describe the flow between the compartments. For example,

here the differential equations are, for concentration $C1$ and $C2$, respectively in compartments $V1$ and $V2$:

$$\frac{dC_1}{dt} = k_{21}C_2 - k_{12}C_1 - k_eC_1,$$
$$\frac{dC_2}{dt} = k_{12}C_1 - k_{21}C_2.$$

5.4 TUCUXI Integration in Clinical Practice

Usually, the pharmacologist receives a list of pending requests for TDM interpretation. He selects a request from the list to analyze the concentration of the drug and evaluate the dosage adjustment called for. For each request he needs to query additional specific information about the corresponding patient. In this Section we describe the process of integration of the TDM software presented in the previous Section into clinical data flow. First, we show how the interoperability was achieved. Second, we present data structure and messages designed to exchange the data.

5.4.1 Interfaces for Data Exchange

In order to integrate the TDM software described in Section 5.3 into the clinical data flow presented above we had to define how to exchange healthcare data between heterogeneous systems that support different data formats. Medical data are stored and exchanged between clinical databases in HL7 format. However, the software operates the data in XML format and produces the reports and graphs in PDF and PNG format. To solve this interoperability issue Mirth Connect [2] – an open source healthcare integration engine – has been used.

[2] https://www.mirth.com

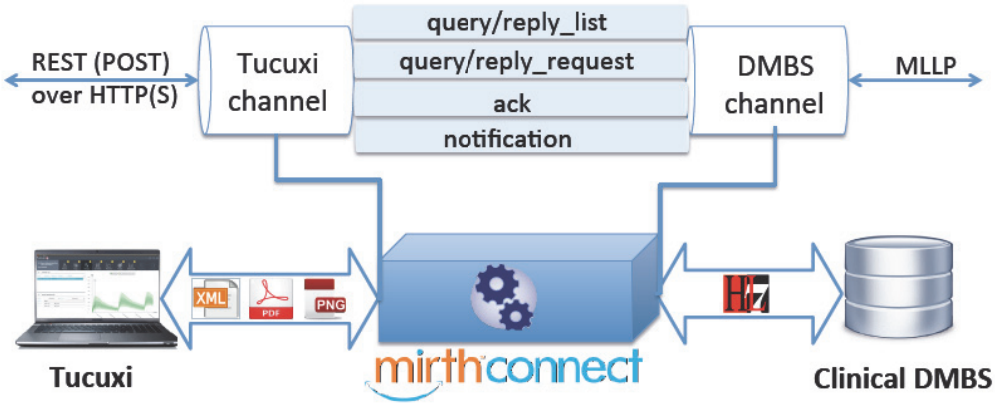

Figure 5.4 – Communication between *TUCUXI* and the database of the medical institution

Figure 5.4 shows the actors involved in the process of data exchange and the formats of the data they use. We use a client-server architecture with a proxy to model communication between *TUCUXI* and clinical database management system (DMBS). We created two external channels for communications: one for *TUCUXI*-Mirth and one for DMBS-Mirth; and four channels were deployed on Mirth for the data transformations. To connect *TUCUXI* to another system there is no need to modify the software, only one Mirth channel may need to be adapted.

Communication between TDM software and proxy is done using a REST API. Sending a query or an update is initiated via an HTTP(S) request that encapsulates the corresponding message in XML format (cf. Figure 5.5). The client *TUCUXI* can send 2 types of query requests, an acknowledgment message (when a response is received) and a notification to update the DMBS. For each type of these messages the separate channel is deployed on the proxy and a message is filtered to the corresponding channel based on the message type. For all the messages sent by *TUCUXI* except the acknowledgment (there is no response for the ACK), the response will go through the same channel and will be transformed from HL7 format to XML. Transformations are defined separately using javaScript for each type of message.

Communication between Mirth and clinical DMBS is defined according to the Minimal Lower Layer Protocol (MLLP) – a standard for transmitting HL7 messages via TCP/IP.

We have implemented the data flow as shown on Figure 5.5. The sequence diagram that describes the data flow consists of the following steps:

1. Obtaining the list of pending requests ;
2. Obtaining the detailed data about specific request ;
3. Individual dosage evaluation, adjustment proposition ;

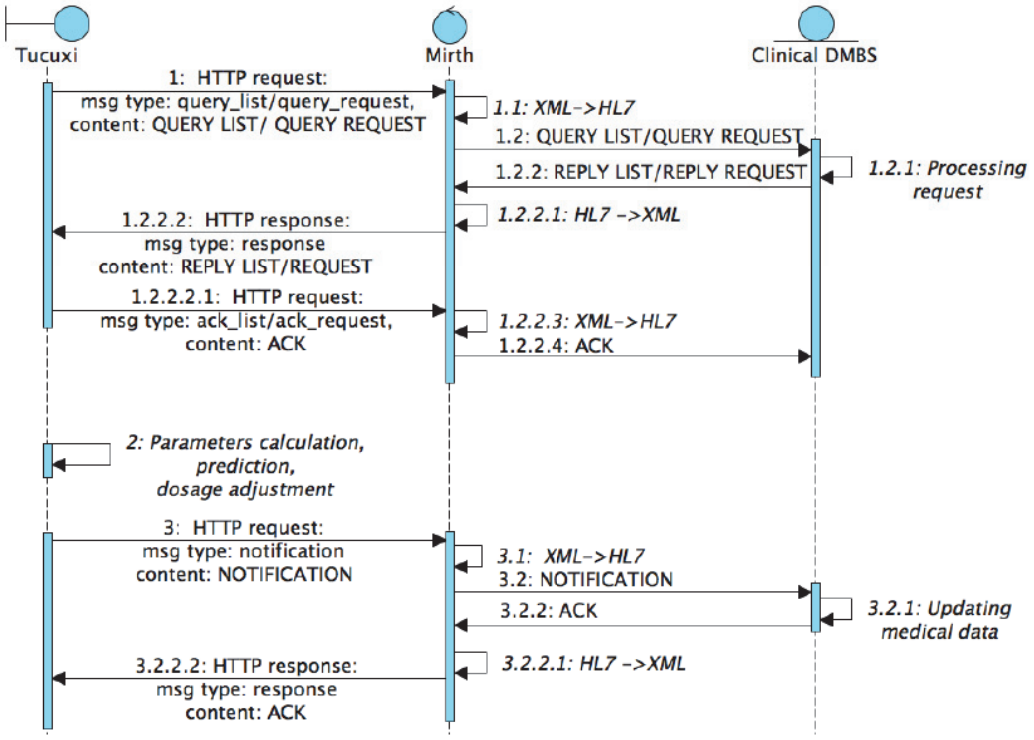

Figure 5.5 – Communication diagram for the clinical data flow with *TUCUXI*

4. Sending results to the medical database.

At the source/destination of these channels we deployed a transformer using javaScript that perform the mapping and construct messages of each type required by the clinical flow. All the messages are received on an external channel that redirects the messages based on the filter defined with respect to the message header where the type of the message is specified.

5.4.2 Data Structure and Messages

In this Section we present the content of the messages we constructed for data exchange between *TUCUXI* and a clinical database. Following the clinical flow of TDM we need to provide the user of the software (doctor, or pharmacologist) with the list of pending requests. A user can specify a time frame that will be an inclusion criteria for the requests based on their arrival date. According to standardized procedures of routine non-automated TDM in the hospital a code "CPCL" with the value "4CPCL" ("0CPCL") is used to tag whether request has (not) been already processed by a pharmacologist. Therefore in order to ask for the list of requests, a `QUERY LIST` message contains the time interval corresponding to the sampling arrival date/time and CPCL value: "0CPCL".

A `REPLY LIST` message is a list of pending requests and it contains, for each request, the

following information:

- request Id – a unique identifier of the request within the clinical system ;
- CPCL code value ;
- information about the patient such as covariates (gender and age), as well as address and the patient identifier in the clinical system ;
- practitioner data: information about the doctor that prescribed the test, and the medical institution, if the sample is sent by another hospital ;
- sample data: identifier of the sample (identifier of the tube form the laboratory), the date/time of the sampling and sample arrival date/time in the laboratory ;
- information about the drug: code used in the system, active principle, brand name, Anatomical Therapeutic Chemical (ATC)[3] classification system code.

This information allows the pharmacologist to ensure that the samples with the medications that require urgent analysis are validated on time.

To obtain more information about the request selected for validation, the `QUERY REQUEST` message is sent to the clinical database system. Multiple drugs can be measured in one sample and several samples can correspond to the same patient. Therefore, a `QUERY REQUEST` contains the patient unique identifier, request id and drug id. This combination uniquely identifies a request for validation of the drug.

`REPLY REQUEST` message in comparison to `QUERY LIST` contains extended information such as the following:

- dosages: start of the treatment or the date of last change of dosage, date/time of the last dose intake, current dosage, frequency of intake, route of administration, comments provided by the clinician ;
- sample results, containing observed concentration analyte and corresponding value and unit ;
- additional patient's covariates required for the drug model: bodyweight, renalfailure (y/n), lastcreatinine, hemodialysis (y/n), hemofiltration (y/n), gestationalage (for newborns), liverfailure (y/n), childpugh, heartfailure (y/n), lungfailure (y/n)), together with the value, unit and the date/time date of the covariate' s acquisition ;
- Timestamped clinical data such as clinical diagnosis, the adverse effects information (toxicity), indication that corresponds to the motivations to TDM, response.

[3] www.whocc.no/atc_ddd_index/

While the `REPLY LIST` message contains the information that can help the clinician to review and choose next request to be validated, extended information from the `REPLY REQUEST` is used to fill in (or pre-fill) the panels (1)-(6) of the software. When a request is processed, and the report is generated, the `NOTIFICATION` message is sent to the clinical database with the following information:

- expectedness: the interpretation of the normality of the result by the analyst ;
- suitability of the treatment: the interpretation of the appropriateness of drug exposure by the analyst ;
- prediction: the recommendation of dosage adjustment by the analyst ;
- remonitoring: the recommendation for future monitoring by the analyst ;
- a priori and a posteriori parameters of the mathematical model ;
- some cautionary statement by the analyst and the timestamp of the interpretation ;
- image of the expected curve (in a binary format in base-64) ;
- report described in Section 5.3 generated automatically ;
- CPCL code with the value "4CPCL", corresponding to the validated request.

If any information about the patient has to be modified (due to a possible mistake originated from clinical database) it can be updated using a `NOTIFICATION` message.

When *TUCUXI* receives a `REPLY LIST` or a `REPLY REQUEST` message an `ACK` message is sent to the clinical database to acknowledge reception of the messages from clinical DBMS. Reception of `NOTIFICATION` is acknowledged by the clinical DMBS as well.

While constructing the messages with the the information listed above in HL7 v2.4 format, custom segments were introduced. Custom segments were used to express status of the request (validated or awaiting validation), information about drug and dosages, some clinical data, the report and the curve with the predicted concentration, as well as a priori and a posteriori parameters of the mathematical model.

While constructing the messages with the the information listed above in HL7[4] v2.4 format, custom segments were introduced. Custom segments were used to express status of the request (validated or awaiting validation), information about drug and dosages, some clinical data, the report and the curve with the predicted concentration, as well as a priori and a posteriori parameters of the mathematical model. For each query/reply pairs we define so-called Conformance Statement – the information that identifies the query, specifies what items can be queried and describes what the response will look like. Table 5.1 shows an example

[4] http://www.hl7.org.

Table 5.1 – Summary of Conformance Statement for the pair of messages `QUERY LIST – REPLY LIST`

Query Statement ID	Z03
Type	Query
Query Name	Request laboratory data
Query Trigger (= MSH-9)	QBP^Z03^QBP_Q11
Query Mode	Both
Response Trigger (= MSH-9)	RSP^Z04^RSP_Z04
Query Characteristics	Selection criteria: the arrival date/time of the request and "CPCL" code
Purpose	To retrieve from clinical DMBS a list of requests for validation for the specified time frame
Response Characteristics	Returns list of requests ordered by arrival date/time
Based on Segment Pattern	RDS_O01

of the part of CS: the table that summarizes the characteristics and identifying information about the query that involves messages pair `QUERY LIST – REPLY LIST`.

5.5 From Drop of Blood to Research Database and Back

Research studies that use retrospective medical data have become a major source of contributions to the biomedical science literature [HMBW13]. Therefore, data aggregation for the research purposes is a essential step towards enhancing clinical literature, e.g., by developing new PKPD models.

Figure 5.6 shows the data flow in a healthcare system spanning from a patient bedside to the healthcare data aggregated in the cloud and used for the research purposes.

Characteristics of the data to be acquired for a research study are determined by the requirements of the study. Therefore, multiple databases need to be constructed. For this we need a system that will connect researchers and medical institutions and will allow them to collaborate with each other and will enable dynamic data aggregation. We assume that the number of data sources participating in the data aggregation should not be static as well as description of the data to be aggregated (could be adjusted during the process of data aggregation depending on the data availability) [DUB+16]. Dynamicity of such system will allow one to accelerate a notoriously time-consuming data collection process.

We developed a mutli-agent system for dynamic data-aggregation for the research purposes. An agent is modeled as an instance of *TUCUXI* used by a doctor or in a hospital. It has been tested using anonymized TDM data and presented in detail in [DUB+16].

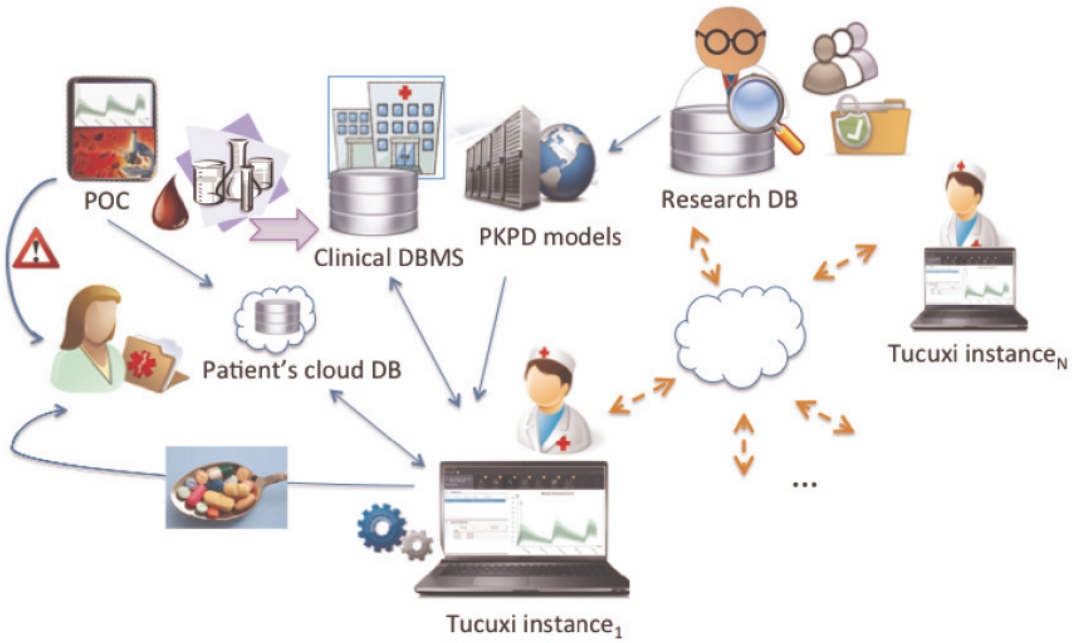

Figure 5.6 – Connecting multiple instances of TDM software

In clinical practice unless the patient provided a consent for sharing the data as they are, the data have to be anonymized. We also developed a privacy-preserving algorithm for distributed data aggregation for medical research [DUV+15] that can be used to ensure anonymity of the patients. However, anonymization may affect the utility of the data. To overcome this issue we are currently evaluating an algorithm that will allow to improve the data utility with the database growth while preserving patient's privacy.

Expanding TDM to a larger patients population by making it available at the Point-of-Care (POC) system will further advance the outcome of drug therapies. Key for widespread dosage adjustment is the availability of point-of-care devices able to measure plasma drug concentration in a simple, automated, and cost – effective fashion. Cappi et al. introduce and test such POC device. The authors present a portable, palm-sized transmission-localized surface plasmon resonance (T-LSPR) setup, comprised of off-the-shelf components and coupled with DNA-based aptamers specific to the antibiotic tobramycin [CSM+15]. Mobile version of the software e.g., running on the POC (cf. Figure 5.6) or on the tablet connected to POC, can be used to alert a patient in case of toxicity of inefficacy of the treatment and provide a recommendation to ask for medical assistance. Dosage adjustment can be made faster as there is no need to send the blood sample to a laboratory and to wait for the results to be transmitted back [CSM+15]. The results from POC could also be sent to the patient's cloud database, accessed by the clinician or pharmacologist and analyzed using TDM software.

5.6 Discussion

In this Section we discuss the ethical issues related to the use of automated software to manage patient's healthcare data.

5.6.1 Ethical Issues

Decision Making in Medical Domain

A piece of software like *TUCUXI* does not aim to replace physicians, but rather to provide assistance in TDM. Dosage individualization is suitable or even required for many drugs, but it is difficult and time consuming, the way it is done at present. Current tools such as Excel worksheets or non-ergonomic softwares are not well suited to manage heterogeneous patients' data in order to issue dosage adjustment decisions on a large scale. Moreover, the time required for a consultation with a clinical pharmacologist may delay important adjustments of the treatment. We think it is possible that clinicians use our software without consulting a specialist in pharmacology. For this we need to ensure "safety" of the software, i.e. its ability to detect unusual cases and to produce reports of no worse quality than those produced by trained clinical pharmacologists.

Before using any PKPD model for a given drug, the model will require approval by a trained pharmacologist after thorough testing. A semi- or fully-automated procedure may contribute to efficient validation of such drug models in the future. Certification of the whole software along with its reference PKPD database is required. Currently we are developing the necessary trials to make it a validated medical device according to applicable regulation.

Evaluation of *TUCUXI* as a TDM Software

A recent review [NCW+17] describes four steps to implement pharmacometrics-based decision support tools, consisting of validating scientific components, defining technical options, considering regulatory aspects, and achieving efficient commercialization. Examples of pharmacometrics-based decision tools that support monitoring of patients and individualization of treatment strategies in neonates, children and adults are presented. Concretely, the evaluation of a medical software such as *TUCUXI* requires several tasks:

1. Verification of the **Correctness of implementation of mathematical models**, for instance, using automated mathematical validation of the software against NONMEM – the *de facto* standard for studies in pharmacokinetics [BSB94].
2. **Validation of data exchange** through a series of scripts aiming to obtain the list of pending TDM requests, run multiple times a day.
3. While the first two tasks are being run automatically every day, thanks to scripting facilities, **clinical validation** requires a real validation in clinical practice. An evaluation

protocol has been designed and is meant to be used after the first two tasks proved the correctness of the software.

Fully Automated TDM

While the GUI version of *TUCUXI* is currently being tested, a next server version is under development. Its goal is to be able to propose an automated interpretation based on all data sent to the server. Basically it is meant to work as the GUI version, but automatically calculating predictions and dosage adjustments. A report will be generated, with graphs and a list of best dosage candidates for the specific patient. This server will allow to be integrated into an electronic patient infrastructure, as a service for analysis labs. Ethically speaking, a human should still be responsible to check the report and to choose the best dosage.

5.6.2 Patient's Privacy

Having access to the population data is crucial for building new mathematical models, or improving characteristics and parameters of the existing ones. According to the current data protection legislation in US[5] and in Europe [Eur16], as well as to the EU General Data Protection Regulation (GDPR) that will replace the European Data Protection Directive 95/46/EC starting from 25 May 2018[6], collecting and sharing personal data require signed consent from the patient to allow using his data for research purposes. Not all patients are willing to provide consent due to the risk of their data misuse [SBH+07, SS16]. For example, if the healthcare data become publicly available insurance companies may infer that a person has a chronic disease susceptibility, and may refuse an application or reject the renewal of their insurance policy. An employer may try to infer healthcare information about potential employees and based on the sensitive information (a serious health condition or a chronic disease susceptibility) may discriminate the candidate [DUB+16]. As an alternative to the consent collection, the data can be anonymized: the patient's data to be used for the research must not be linked to the identity of a person to whom these data belong [EERM15, Eur16].

How to ensure that the data that may belong to the same patient and were aggregated from different sources were properly anonymized? In [DUV+15] the authors proposed a privacy-preserving algorithm for independent release of medical data in distributed environment. However, one should take into account that absolute anonymization is only possible when no data are shared at all and, therefore, the privacy-utility trade-off needs to be found for every specific case. How to automatically adapt this trade-off for different databases? What is an acceptable risk of violation of patient's privacy? These questions still remain open.

[5] http://www.hhs.gov/hipaa/
[6] http://www.eugdpr.org/eugdpr.org.html

6 Data Aggregation for Precision Medicine

The collection of medical data for research purposes is a challenging and long-lasting process. In an effort to accelerate and facilitate this process we propose a new framework for dynamic aggregation of medical data from distributed sources. We use an agent-based coordination between medical and research institutions. Our system employs principles of peer-to-peer network organization and coordination models to search over already constructed distributed databases and to identify the potential contributors when a new database has to be built. Our framework takes into account both the requirements of a research study and current data availability. This leads to better definition of database characteristics such as the schema, content, and the privacy parameters. We show that this approach enables a more efficient way to collect data for medical research.

6.1 Introduction

Research studies that use retrospective medical data have become a major source of contributions to the biomedical science literature [HMBW13]. Clinical data repositories are promising resources for the development of personalized medicine, clinical trials, epidemiology and public health [WHSW13]. Unfortunately, the collection of medical data is notoriously time-consuming. Data collection in one medical institution may take several years [FGG+14]. In order to accelerate this process, or when required data are diverse and can not be collected on site, multiple medical institutions may collaborate to aggregate the data. However, distributed medical data aggregation is challenging as it requires solving privacy and data quality issues, as well as enabling interoperability between medical systems.

According to the data protection legislation in Europe and US, collecting and sharing personal data requires signed consent from the patient to allow using data for research purposes [VBC+08, CM03]. Not all patients are willing to provide a consent because of the sensitive nature of their medical data. For example, if the data become publicly available insurance companies may infer that a person is suffering from a chronic disease and may refuse an application or reject the renewal of her insurance policy. An employer may try to infer healthcare

information about potential employees and based on the sensitive information (a serious health condition or a chronic disease susceptibility) may discriminate the candidate.

As an alternative to the consent collection, the data can be anonymized to be used in clinical research [VBC+08, CM03]. This could be done by applying existing privacy protection mechanisms [EII+10, XC14, CJ05, SMBMS13]. However, mobility of the patients and a will or sometimes a necessity to visit more than one medical institution can introduce another privacy threat. It has been shown that in the case of the independent release of locally anonymized datasets that contain information about the same patients, their re-identification is still possible (e.g., in the case of a composition attack first described in [BLLW11]). In order to counter these privacy threats, several models in the area of distributed privacy-preserving data publishing have already been proposed (i.e., pseudonymization [EII+10, XC14], secure multi-party computations [CJ05], microaggregation [SMBMS13], cloning [BLLW11]). However, those models can significantly affect the quality and, therefore, the utility of data, since they do not take into account data availability, content, structure and representation.

Both the structure and the representation of the health data that need to be aggregated for the research purposes depend on the requirements of a study. Therefore, it is not possible to specify a unique static schema of the database that will fit different clinical studies. In order to guarantee the data utility and patients' privacy, the database schema and privacy parameters have to be adjusted based on the clinical study, for which the database will be employed.

Building multiple databases for different research studies is for example particularly relevant to one of the key concepts of personalized medicine - therapeutic drug monitoring (TDM) [GWM+12]. TDM transformed drug therapy by providing the ability to characterize sources of variability in drug disposition and response to individualize drug dosing [MW14]. TDM is based on models that allow to compute characteristics of a particular drug based on the patient' covariates. In order to build these models, population healthcare data are needed. The data requirements vary for different drugs and populations (e.g., neonates or adults), and therefore multiple databases need to be constructed.

Our goal is to develop a system that will connect researchers and medical institutions and will allow them to collaborate with each other. This chapter presents a multi-agent system (MAS) for dynamic data aggregation in medical research. We use agents as the problem requires a distributed and autonomous system, where participants can join the network and decide what to search for and what to share independently from the other participants of the network. The participants do not necessarily know each other, and may use different ways to structure their data. By representing participants as autonomous agents in a distributed network, we can then focus on defining all the mechanisms for coordinating the participants to find each other and to share the data in a meaningful way. The system (i) enables the connection of research and medical institutions into a peer-to-peer (P2P) network and (ii) provides an environment to negotiate and define the characteristics of the database such as schema, content, and privacy parameters based on the data requirements and availability.

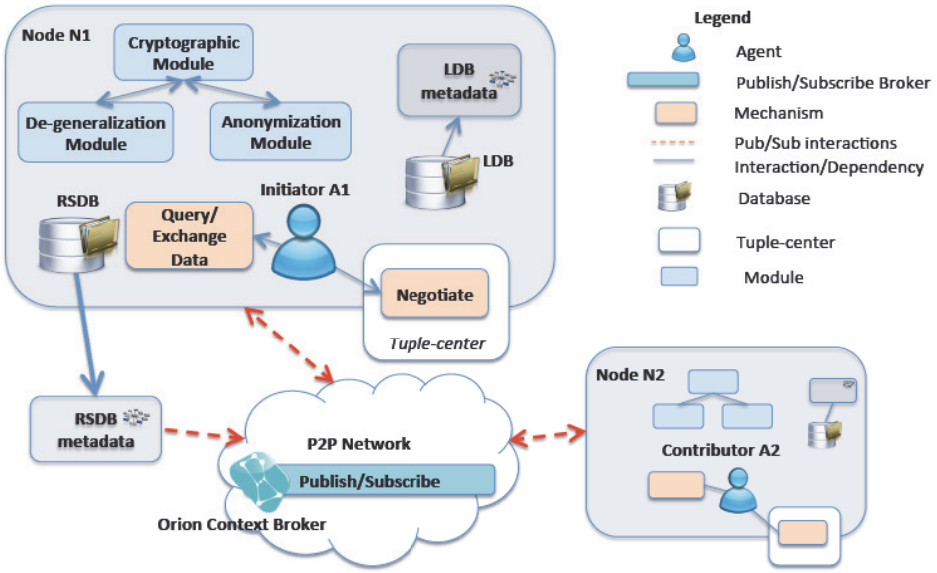

Figure 6.1 – Architecture of the multi-agent system

We evaluate our system using patients data collected in the neonatal intensive care unit over 5 years within the frame of a routine TDM program [FGG+14]. The advantages of our solution are the following:

- A research study can be conducted faster, as the time needed to aggregate the required amount of data is dramatically decreased in the case of using our system with respect to the time needed for data collection in a medical center.
- Multiple databases (satisfying the requirements of different research studies) can be shared between the users of the proposed system: medical and research institutions.
- The sensitive nature of medical data is considered during every step of data aggregation in order to achieve trade-off between privacy and utility.
- The system is "fair" in the following sense: if all users participate in data aggregation, every user will be able to gain access to approximately the same amount of data as he contributes. It means that every user of the system can benefit from the data collection. We believe that this will motivate medical and research institutions to join the system and participate in data aggregation.

The rest of this chapter is organized as follows. In Section 6.2, we provide a use-case scenario and a general description of our framework. In Section 6.3, we demonstrate *dynamicity* of

our system: we present in detail the process of P2P network organization and the agents' negotiation phase. We also provide the necessary background about existing coordination models we build our negotiation mechanism on. In Section 6.4, we discuss some privacy and security concerns. We provide the description of the implementation and evaluation results of the system in Section 6.5.

6.2 MAS Framework

In this section, we show how our system could be used by medical and research institutions. We also present the architecture of the system and describe functionalities of its elements.

6.2.1 Use-Case Scenario

There is growing interest and a strong need to share individual patient data for secondary purposes, particularly for research [EERM15]. The system presented in this chapter will facilitate and accelerate the data sharing and aggregation. We assume the following scenario. Users of the system are research institutions and medical doctors or healthcare institutions that possess the medical data. Users may have the following goals: *(i)* to access anonymized medical data and use them in particular research study, *(ii)* to contribute to the development of research by sharing patient data. For simplicity we assume that there is no economical competition between different research and medical institution.

6.2.2 MAS Architecture

Figure 6.1 presents an architecture of our multi-agent system for dynamic data aggregation, its components, and their relationships to each other and to the environment. The system consists of a publish/subscribe broker that serves as a lookup system, and the nodes that represent users of the system. Based on the user's requirements, one or several *agents* could be initialized by the node. *Agents* are used at different stages of the process of building a research database (*RSDB*) from distributed local sources (*LDB*). First, to find the contributors to the database and, second, to adjust the structure and representation of the data depending on the requirements of a particular research question, current availability of the data, and privacy considerations. These steps require coordinating the participants, interactions and reasoning, hence we employ an agent-based approach.

An *RSDB* is a database with anonymized data to be used for research purposes. Each *RSDB* will be constructed taking into account the requirements of a particular research study, e.g., in case of TDM this could be the concentration measurements of a specific drug in the patient's blood. The information about already constructed *RSDB* (metadata of the *RSDB*) will be shared within the network, therefore, there is no need to aggregate the data again if a similar research study has to be conducted. A user will be notified if there exists a database that satisfies the user's

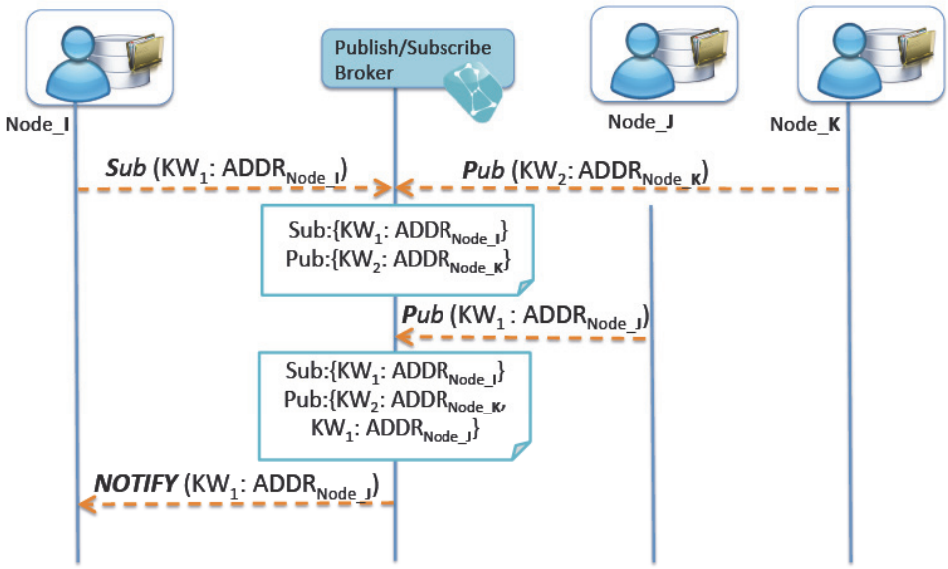

Figure 6.2 – Process of peer-to-peer network organization

requirements.

LDB contains patients' data collected in a medical center. This information will only be aggregated after coordination, agreement on the characteristics of the database and applying privacy and security mechanisms. Metadata of *LDB* consist of the information that describes medical data stored in *LDB* and used to identify the potential sources for an *RSDB*. No sensitive information can be shared during organization of P2P network and agents coordination.

The nodes can interact with publish/subscribe broker to either publish the availability of the data, or make a subscription based on the requirements of a research study. If a new database has to be constructed, we need to identify the sources of the data and to connect them. For this, we use the publish/subscribe paradigm to discover the nodes with relevant data instead of multicasting a request. More information about the process of P2P-network organization is provided in the next section. Nodes can access their *LDB*s and can use the functionality of the following mechanisms: *Query/Exchange Data* and *Negotiate*. *Query/Exchange Data* is used to publish and subscribe using the broker, to query and exchange the metadata, and to transfer the data to the *RSDB*. *Negotiation mechanism* is based on the TuCSoN coordination model [NVP10] and aims at adjusting the characteristics of *RSDB* (e.g., the schema of the database, required number of records to be collected, ang privacy parameters). We focus on coordination between agents in the next section. As a part of the negotiation process, a semantic agreement between schema of different databases and different data representations needs to be established. This is out of the scope of our work, we assume that existing ontologies

and schema matching solutions [NLM+13, NDNTA14] can be employed.

To ensure the authenticity, integrity and anonymity of the data that are being aggregated, we develop the following modules: Cryptographic Module, Anonymization Module and De-generalization module. The functionalities of the Cryptographic Module are (i) to create pseudonyms with which the data about the patient will be uploaded to *RSDB* and (ii) to generate the signature before data transfer in order to ensure the authenticity and integrity of the data. The Anonymization Module uses generalization algorithms that allow to replace the exact values of the data with a range within which these data fall. This guarantees the k-*anonymity* property in a distributed environment, hence ensures the data privacy. De-generalization module is employed to improve the quality of the data with the growth of *RSDB* by mitigating the data losses due to applying anonymization algorithms. The algorithm for data aggregation and the de-generalization protocol are described in detail in Chapter 4.

The dynamics of a system is characterized by constant change, activity, or progress[1]. The term *dynamicity* in the context of complex open and distributed systems can be intuitively defined as the ability for a system to be configured, developed, maintained, modified at runtime, without compromising its integrity and ongoing processes [JH03]. We use the term dynamicity in the following sense. First, we assume that the number of agents participating in the data aggregation is not static: i.e., an agent can join and leave the network. Second, we use term dynamicity to specify that there is no need to have fixed static description of the data to be aggregated. It can be adjusted during negotiation phase. Dynamicity enabled us to accelerate data collection process. Hereafter, we describe two main interaction processes: a publish/subscribe mechanism that helps agents to get organized in a P2P network (Section 6.3.1), and negotiation: a process that enables agents to find an agreement on the data representation, as well as the security and privacy parameters (Subsection **??**).

6.3 Dynamicity of MAS

6.3.1 P2P Network Organization

We use a publish/subscribe paradigm to organize the nodes in the P2P network. It allows delivery of the data from their producers (publishers) to their consumers (subscribers) in the distributed environment in a decoupled fashion [GSAA04]. This means that publishers can introduce the data into the system (publish/subscribe broker) being unaware of the subscribers. Subscribers can register their interests by subscriptions, which filter relevant events to the subscribers. The broker enables publication of context information by publishers, so that the relevant information becomes available to subscribers.

The role of a publish/subscribe broker in our system is to support dynamicity and to allow the node (i) to register availability of a certain kind of medical information within the network; (ii)

[1] http://www.oxforddictionaries.com/definition/english/dynamic

to subscribe for a notification if a certain type of information has been published. This is done to avoid performing active discovery of peers or forcing the publishing nodes to broadcast the network to demonstrate data availability each time when there are new peers join the network. Figure 6.2 illustrates the P2P-network organization mechanism used in our framework. It shows how we structure the messages that are used during the interactions between the broker and the nodes.

After registering at the broker the node subscribes to a certain type of data by specifying a set of keywords ($KW = [kw_1, ..., kw_n]$) that describe the data the node is interested in. Similarly, for a node that possesses the data, it is sufficient to publish the description of the data using the keywords. If the keywords match corresponding subscriber will be notified by the broker and provided with the list of the addresses of the nodes with relevant data available. The semantic description of the data has to be provided by the users of the system. This is why we have chosen a simple keyword approach. In the future work we plan to improve the mechanism for P2P network organization.

6.3.2 Agents Negotiation

Negotiation is a process initiated by a node in order to obtain a certain number of medical records to build an *RSDB* for a particular research study. It is followed by discovery of the nodes with the relevant data. Negotiation is built on interactions between the agents within the TuCSoN coordination model [NVP10]; the coordination is happening through tuple centers (TCs). TCs can be seen as a shared system such as blackboard system [Gel85], where the information is being exchanged in the form of tuples. The templates of the tuples are specified with respect to their structure. An ontology model could also be employed to interpret the information transferred by the tuples. Next, we describe the structure of the tuples at different states of the negotiation.

Using the tuple center, an agent can, for instance, write (`out` operation), read (`rd` operation) or consume (`in` operation) the tuples. Figure 6.3 presents a state diagram for the negotiation process proposed in this chapter and implemented as a part of our framework. The negotiation process demonstrates the states and the transitions between them.

The node that initiates the process of data collection creates a Negotiating Agent (host) to start the process of negotiation. Next, the host creates a TC within its own node, where the negotiation will take place. Once the TC has been successfully created, the agent injects a script that controls the state of the negotiation. The script is written using the first-order logic language ReSpecT [DNO98] and allows to program the behavior of a TC. The following states are possible during the negotiation process:

- **Started**: The host writes into the TC a tuple with respect to the predefined template that consists of a list of agents ($t_1 =$ `invited`$(AgentList)$) that will be invited to take part in the negotiation. When an agent from the list arrives to the TC, it writes a tuple

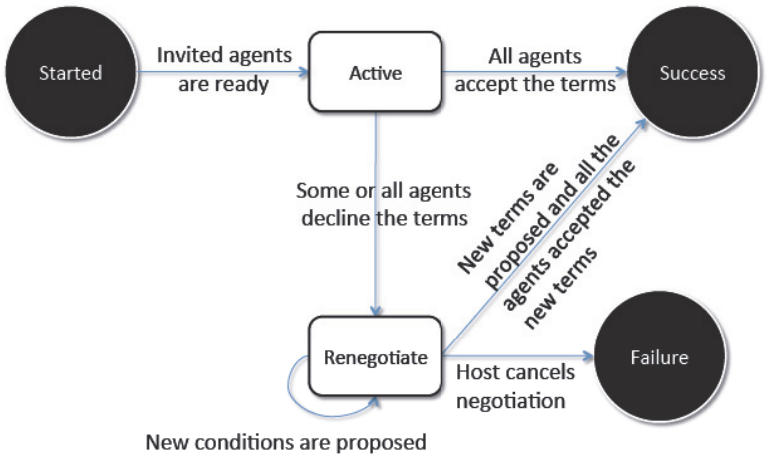

Figure 6.3 – States of the negotiation process

$t_2 = \texttt{hello}(AgentId)$.

- **Active**: When all the agents write the tuple t_2, a reaction that sets the state to **Active** is triggered. At this stage, the node proposes the conditions of the negotiation and the peers evaluate them using the rules. In the case of building the database, the conditions could be the schema of the database (attributes and their ranges), number of records needed (N), as well as privacy parameters if required. A node writes the following tuple specifying m attributes ($attr$), number of records N and keywords:

$$t_3 = \texttt{parameters}(KW, \{(attr_1, min_1, max_1), ..., (attr_m, min_m, max_m)\}, N). \quad (6.1)$$

Then, the conditions have to be evaluated by the other nodes using rules, e.g., the agent J reads the tuple from the agent I and evaluates it as follows:

$$\left(KW^I \subset KW^J\right) \wedge \left[\left(attr_p^I = attr_p^J\right) \wedge \left(min_p^J \leq min_p^I\right) \wedge \left(max_p^J \geq max_p^I\right),\ p \in [1, m]\right] \wedge \left(N^J \geq 0\right). \quad (6.2)$$

If the conditions are satisfied then the agent will write the following tuple:

$$t_4 = \texttt{answer}(AgentId, \tilde{N}), \quad (6.3)$$

where $\tilde{N}$ is a number of records an agent (with corresponding $AgentId$) can contribute to the *RSDB*. We provide an example of the tuples and conditions that we used during evaluation in Section 6.5. A threshold for the peers to respond is used to bound the maximum duration of this state.

- **Renegotiate/Failure:** If the conditions of data exchange proposed by the host are not accepted by one or more peers, it is possible to either terminate the negotiation by setting it into the **Failure** state and marking the TC as reusable or to set the state to **Renegotiate**. At this state the list of participants could be changed, and the peers can modify the parameters of the tuples. Currently, acceptance of the terms is based on the user engagement. When the **Failure** state is reached, all agents terminate.
- **Success**: If the terms are accepted by all the peer agents, the data transfer occurs. When each node finishes data transfer to the host, the host marks the agent as finished writing a tuple $t_5 = \texttt{finishedAgent}(AgentId)$. When all the agents from the invite list have been marked as finished, **Success** state is triggered, effectively ending the negotiation process as all agents terminate when this state is reached.

In the end of this process, the host agent will either obtain a sufficient amount of data or it will be waiting for other (or existing) peers to join the negotiation again to complete aggregation of data. The host will be notified by the broker if an agent publishes at the publish/subscribe broker information about the availability of the data. Then the agent will be able to join the negotiation process. The state will return to **Active**, repeating this cycle until the host obtains the desired amount of data.

For the sake of simplicity we do not present the structure of all the tuples that we use to model the reactions in the cases such as removing an agent from an active negotiation process, or changing the status of the negotiation process.

6.4 Data Security and Privacy

Hereafter we discuss privacy and security requirements to the medical data before they could be transferred in the case of distributed data aggregation for the research purposes. We also describe how we are going to address the need for privacy and utility trade-off in our system.

6.4.1 Need for Security and Privacy

In order to be sure that the research database contains only veritable medical data, it is very important to provide integrity and authenticity of the data, i.e., to insure that the data are correct, the data source is a real medical institution and that it is possible to re-contact the doctor that provided the data (if needed). Therefore, the certification authority needs to be deployed and every time the data are sent to the research database the use of digital signature [GD09] is required. These methods are standardized, and their functionality can be provided through the cryptographic module at every node.

As already mentioned it is impossible to have one fixed data structure for different types of medical research. Therefore, privacy preserving mechanisms need to be adapted for different datasets. In [Swe02] authors proposed the notion of k-anonymity: ensuring privacy by con-

structing a set of k records indistinguishable in terms of QID – quasi-identifiers a set of the attributes that can (in combination) identify a person. This approach is based on applying generalization functions to QID and suppression to uniquely identifiable patients data. K-anonymity guarantees that the probability to de-identify a person to whom a record belongs does not exceed $\frac{1}{k}$, where k-is the cardinality of the set of indistinguishable records.

6.4.2 Privacy-Utility Trade-off

Anonymization certainly affects the data utility [CSLL10]. Therefore, it is of a hight importance to be able to minimize the anonymization while preserving the patient's privacy. To achieve this, we propose to adapt privacy parameters taking into account the format of the data that will be collected. The utility expectations should be specified depending on the requirements of a particular research question. And this will be the basis for defining privacy parameters and the generalization functions for each of the attributes.

In our MAS the values of the privacy parameters can be seen as one of the conditions specified by host based on the utility expectations. Every contributor can propose to modify the parameters during the process of agents negotiation described in Section 6.3. In Chapter 4 we proposed an algorithm that allows to release medical data for the research purposes from different *LDB*s independently, while preserving the anonymity property of RSDB and improving data utility with the database' growth. Generalization rules are expressed as binary trees and are used to achieve k-anonymity and maximum utility without revealing non-anonymized QID values to the system. The algorithm also employs pseudonymization technique and provides a possibility to recontact the patient through a caregiver that uploads the data. This functionality can be used by an agent that can now employ proposed anonymization algorithm before making a contribution to the *RSDB*.

6.4.3 Data Transfer

Before the data are transferred the anonymization algorithms [DUV+15] are applied. This guarantees that k-anonymity of RSDB is preserved, and, therefore, patient privacy will not be violated. The data are transfered using a separate web service. When a new RSDB is constructed, its metadata are sent to the broker and are kept updated. This allows one to reuse the database if needed, or populate it with more records, keeping the data consistent and private.

6.5 Results and Discussion

In this section, we provide the details about development and virtualization environment that has been built in order to implement the MAS described above. We describe the datasets that have been used to evaluate the consistency, performance and the scalability of the system.

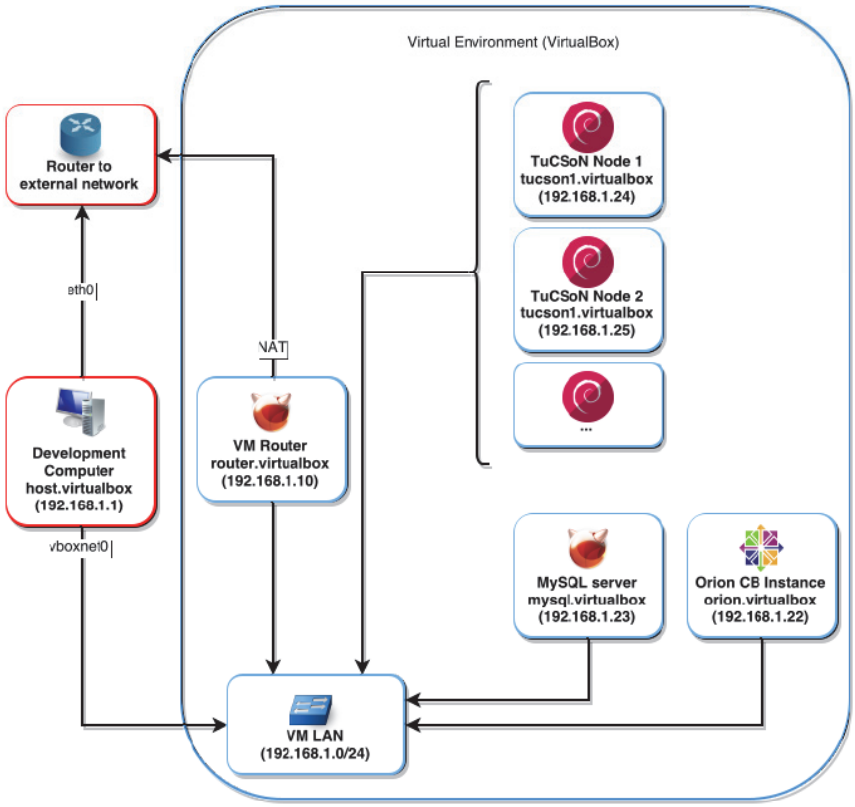

Figure 6.4 – Virtualization environment

The results of the evaluation are discussed in Section 6.5.4.

6.5.1 Development and Virtualization Environment

For the system development, the Java language has been chosen based on the following reasons: the programming API of TuCSoN is written in Java, a high-level language is required to program the complex tasks the agents perform, and the execution is reasonably fast. The machine used for development runs GNU/Linux, specifically, Xubuntu 14.04. The system runs on top of a VT-x capable Intel Xeon CPU with 8 logical cores and 16 GB of RAM.

To test the system, we set up a virtualized environment with VirtualBox 5.0.1 used as a virtualization engine. As shown in Figure 6.4, the virtualization environment is comprised of several virtual machines and a host-only network, which isolates the virtual machines and the external network environment to avoid unsolicited traffic interfering with the virtualization environment. Outbound access from this network is routed through a virtual machine hosting a DHCP server and DNS server. All the virtual machines were running with a KVM-compliant Paravirtualization layer and Hardware-assisted virtualization through Intel VT-x. Virtual machines running FreeBSD as guest OS have a Hyper-V layer instead. Table 6.1 illustrates the setup of the virtualization environment.

Table 6.1 – Functionality and characteristics of virtual machines

Name	Functionality	Guest OS	CPU, cores	RAM	Disk
VM Router (router.virtualbox)	Routing traffic from the Virtualization; Environment to the Internet; Hosting a DHCP server and DNS server	FreeBSD 10.1 x86	1	512 MB	8 GB
Orion Context Broker Instance (orion.virtualbox)	Hosting an instance of the Orion Context Broker	CentOS 6 amd64/ RHEL 6 amd64	2	4 GB	20 GB
TuCSoN Node (tucsonX.virtualbox)	Representing a Node in the network (also requires JRE 8 to run Java code)	Debian 8 amd64	2	1 GB	8 GB
Database (mysql.virtualbox)	Acting as a MySQL server as a storage backend for medical data	FreeBSD 10.1 amd64	2	2 GB	20 GB

Table 6.1 shows the functionality and characteristics of each virtual machine used in the implementation. One has to notice that for the evaluation we deployed a single MySQL instance into which each node operates using its own database. In a real-life scenario, each node would have its own storage backend located at each node.

6.5.2 Dataset

The dataset is comprised of two separate databases, one with 8898 records (called *Gentamicin_large*) and a second one with the extended schema, containing more health information within 224 records (called *Gentamicin_small*), in total, the database consists of 9122 medical entries. The data has been collected in pre-term and term newborns treated with *Gentamicin* (an antibiotic) in the neonatal intensive care unit at CHUV and has been used both for the treatment and for the research purposes in the framework of the ISyPeM project[2]. The data had been previously statically anonymized in the hospital, such that it is impossible to de-identify patients. We consider the following attributes of a record: a pseudonym of the patient, body weight, gestational age, postnatal age, gender and various information related to the concentration measurements of an antibiotic in the patient's blood. Based on the semantic of the data we annotated the dataset with the following keywords: "Gentamicin" and, "neonates". The following attributes have been chosen: body weight (BW), gestational age (GA), postnatal age (PNA), gender, and concentration of the drug in the blood. We discarded some records that have missing values corresponding to any of the attributes listed above. This reduced the size of the resulting dataset to 8922 records.

[2]http://www.nano-tera.ch/projects/368.php

To diversify subscriptions and the data that the nodes have we added some synthetic datasets annotated with the keywords “Malaria”, “adults”, “cancer” with the attributes age and gender.

6.5.3 Evaluation Scenario

Our solution extends the work done by Urovi et al. in [UOdlT+14, UOB+12] by providing a way to collect the data about different patients from multiple sources, and anonymize the patient’s data so that, even if records are shared in *RSDB*, the patients’ privacy is preserved. The dynamic creation of *RSDB* was out of the scope in [UOdlT+14, UOB+12]. In addition, we define a negotiation process for which these data can be aggregated dynamically. Nonetheless, the work of Urovi et al. shares some of similarities to our own, notably the use of TuCSoN coordination model [NVP10] for the agents-negotiation phase.

The system presented in this chapter is also close to the approach for sharing the data proposed in [WVNL14], however, there are the following important differences. Negotiation phase of our solution, preceding the actual data exchange step, enables the nodes to agree on the common schema for a particular database (instead of managing multiple schemas from different local servers). In our solution, we minimize the use of centralized approach, by only employing it for P2P-network organization (in contrast to sharing schemas through the central server as in [WVNL14]). Therefore, if the broker is temporally overcharged and is not available, the peers can continue the data aggregation process within P2P networks that have been already organized.

To the best of our knowledge, there is no system to benchmark with since existing systems do not provide the same functionality or work in different environments. Therefore, we proposed the following evaluation scenario. We first test the consistency, performance and the scalability of our system. Second, we prove our initial assumption that the system is “fair” ; this means that an agent that participates as a data provider can also obtain the data it needs. And the more the system is used, the closer to the equality is the amount of data an agent could provide and obtain.

We defined a set of 20 hardcoded conditions that differ from each other in values and combinations of parameters such as body weight, gestational age and gender. For example: *({“Gentamicin”, “neonates”}, {(“BW”, 2000, 3000), (“GA”, 38, 42), (“gender”, any), (“concentration”, any)}, 6000)* expresses the conditions for the dataset that aims at having 6000 records about neonates with bodyweight between 2000 g and 3000 g, gestational age from 38 to 42 weeks, and any gender and any concentration value.

To test consistency, we compare the results of using the same condition (selected randomly from the predefined set) in the case of querying the database directly (equally to 1 agent, or to a local database) and in the case when the data are distributed between 3, 5, and 10 agents. We make a realistic assumption that the number of participants for populating one database would rather not exceed 10. However the number of data publishers is not limited by our

system. We assume that there is always one agent that acts as a subscriber and all the other agents are publishers. The subscriber can also possess the data and make a contribution to the database. We evaluate performance and scalability by measuring the time of a system run, $t^{\text{run}}(n)$, for different number of agents, n, that are ready to provide the data. We consider the running time as a time between the moment when subscriber in P2P network receives the notification about the data available and the moment of the dataset creation.

We evaluate "fairness" of our system by estimating "gain" and "loss" for every agent, participating in the data exchange for the different number of agents. We simulate the settings in which every agent randomly selects a condition from the predefined set and initiates the process of a dataset creation. We split the data randomly between 10 nodes: we populate each node's database with approximately 800 records. We then calculate an average difference between the number of records obtained and the number of records provided by a single agent when using our system after different number of runs. To avoid contingency we averaged out the results over all the nodes participating in the data exchange.

6.5.4 Evaluation Results

Table 6.2 – Evaluation of performance and scalability

Number of agents, n	1	3	5	10
Time of a system run, $t^{\text{run}}(n)$, sec	1.3	21.6	25.6	46.1
Time of local data collection, t^{loc}, months	60	-	-	-
Time of distributed data collection, $t^{\text{dist}}(n)$, months	-	20	12	6

As expected for each condition, the number of records obtained from the databases from distributed sources, including the database of the initiator, always sums up to the number of cases obtained from the querying database before splitting the data. Therefore there is no data loss and the system is consistent.

The results obtained while evaluating scalability and performance are presented in Table 6.2. Table 6.2 shows that the system is scalable, yet the time of the system run increases with the number of agents, it does not exceed one minute in the case of 10 agents. Important notice is that before aggregation is possible, the data have to be available locally: already collected by a medical center. Nevertheless, our system significantly decreases the amount of time needed to collect the required amount of data. Hereafter we compare the time of data collection performed entirely on site (in one medical institution) with the time needed to collect the same amount of data using our system that allows to connect n different medical institutions. We also discuss the results presented in Table 6.2.

The required amount of data to be collected for a specific research question can be expressed as a number of records, corresponding to different patients, or in case of TDM as a number

of concentration measurements of a specific drug in the patient's blood.[3] Let's assume that we are interested in obtaining D measurements. For simplicity, we assume that each medical center or laboratory performs at least some certain number of tests per months, r. Then, the time t^{loc} required to collect D measurements in one medical institution can be expressed as

$$t^{\text{loc}} = \frac{D}{r}. \tag{6.4}$$

If we have an access to multiple data sources (n local databases) then during one months there will be $n \times r$ tests available. Therefore, we can define the time needed to obtain D measurements from n databases, $t^{\text{dist}}(n)$ (taking into account the time of a system run, $t^{\text{run}}(n)$). Then we can compare it with the time t^{loc} needed to collect the same amount of measurements in one medical institution.

$$t^{\text{dist}}(n) = \frac{D}{n \times r} + t^{\text{run}}(n). \tag{6.5}$$

$$\frac{t^{\text{loc}}}{t^{\text{dist}}(n)} = \frac{D}{r} \div \left(\frac{D}{n \times r} + t^{\text{run}}(n) \right). \tag{6.6}$$

Local data-collection usually requires months, but as Table 6.2 shows the time of a system run, $t^{\text{run}}(n)$, does not exceed a minute for up to 10 agents ($n = 10$). This enables us to simplify equation (6), as $t^{\text{run}}(n)$ is negligible compare to t^{loc}:

$$t^{\text{dist}}(n) \approx \frac{t^{\text{loc}}}{n}. \tag{6.7}$$

Equation (7) shows that the time required for distributed data aggregation using our system, $t^{\text{dist}}(n)$, is approximately n times less then the time t^{loc}, needed for on-site collection of the same amount of data, D. For example, for the dataset we used for the evaluation collection of the data in one medical institution took approximately five years [FGG+14]. Using 10 sources of data, for instance, enables one to collect approximately the same amount of information we used for the evaluation during a half a year instead of five.

To show the "fairness" of the system the results of the simulations with the 10 agents setup are shown on Figure 6.5. We measured the difference between "gain" and "loss" for every agent for the increasing number of runs. Negative values indicate that after a number of runs an agent provided more records then it obtained, while positive values show the opposite. We noticed that some nodes do obtain or do provide more cases than others, but on average the difference is low. Furthermore, we can see that the average difference between the number of records provided and the number of records obtained during the use of the system decreases with the increasing number of runs. Therefore, the more time the system is in use, the closer it

[3] If we consider different medical records we should take into account that the information about the same patient can be stored in multiple databases. To avoid multiple entries in the *RSDB* corresponding to the same patient, Cryptographic and Anonymization modules have to be used.

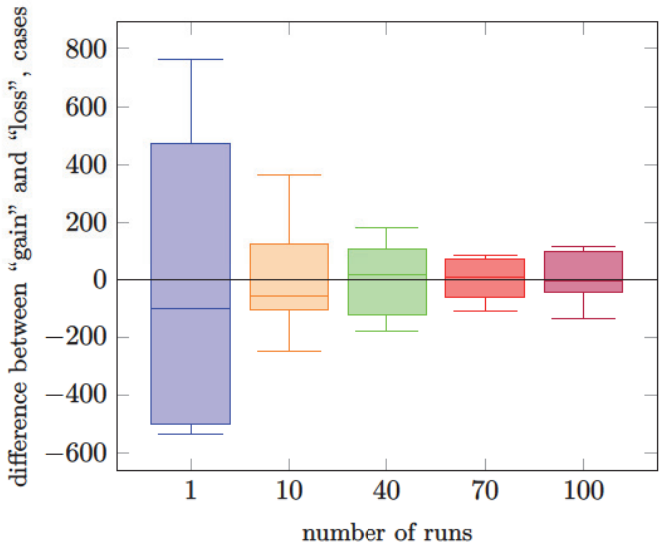

Figure 6.5 – Simulations
The graph shows how the difference between the amount of data provided and obtained by an agent changes with the increasing number of system runs

is to a "fair" state, i.e. when the difference between the number of records provided and the number of records obtained by an agent converges to zero.

7 Concluding Remarks

In this thesis, we have addresses important challenges in the context of eHealth: providing privacy and achieving interoperability. However, the solutions presented in this thesis could also be applied in other contexts, where the sharing or aggregation of any sensitive microdata is required. These solutions are relevant particularly for cases when the information about a same individual is present in multiple databases. In this final chapter, we summarize our contributions and present potential extensions of this work.

7.1 Summary of the Thesis

In the first part of the thesis, we addressed the privacy challenge. After studying from practical and legal perspectives requirements and regulations in the framework of privacy-preserving healthcare-data management, we focused on the two components required to guarantee a patient's privacy in the context of sharing healthcare data: the patient's control over his personal data and the data anonymity guarantees in the case of data de-identification.

In order to ensure that the patient has control over his healthcare information, even when the data are shared among multiple entities, and to ensure that the patient can define and reinforce access control policy, we proposed to use blockchain technology. In healthcare, a distributed ledger can be seen as a shared immutable and transparent history of all the actions performed by eHeath users; these actions including defining access control policies, sharing, accessing and modifying the data.

We assessed the benefits that blockchain technology could bring to primary care, medical research, and connected health. We reviewed the existing implementations and, using the constrains related to the healthcare context, we explained our choice of the permissionned blockchain technology for the implementation of the proposed scenarios. We also presented an architecture of the framework for the specific needs in the case of radiation oncology data-sharing, and implemented a prototype that ensures privacy, security, availability, and fine-grained access control over highly sensitive patient-data.

Tackling the problems of distributed medical data de-identification, such as risks of violation of the data anonymity, limited utility of the anonymized data, dynamicity of the patient data, we proposed a framework for secure and scalable privacy-preserving utility-aware data aggregation. We used anonymization and pseudonymization approaches to develop an algorithm that constructs an anonymized database with patient data for medical research purposes, and provides a possibility to continuously update the database. Then, we designed a privacy-preserving protocol that improves data utility by de-generalizing the database records in a distributed environment. To achieve privacy, we employed an efficient functional encryption and the Shamir secret-sharing scheme. We implemented and evaluated our utility-improvement solution by using a benchmark dataset.

In the second part of this thesis, we addressed the challenge of achieving interoperability in healthcare. This work was done in the framework of therapeutic drug monitoring, which enables healthcare professionals to individualize the doses and helps them to ensure the best possible outcome for each patient. We integrated the *TUCUXI*– recently developed existing TDM software – into a clinical practice. This required studying and adjusting the clinical-data flow, as well as designing and implementing interfaces and messages to achieve interoperability with the clinical database management system. We also analyzed and discussed the challenges and ethical issues related to the automation of therapeutic drug monitoring in patient care and medical research.

Then, to provide a possibility for aggregating data that are collected and produced by the TDM software *TUCUXI*, we developed and implemented a multi-agent system for dynamic data aggregation. The system enables us to facilitate and accelerate the process of data aggregation and to build a research database with the possibility to update it dynamically and to preserve the patients' privacy. The data-aggregation mechanism can be adapted on the fly based on the research study requirements. The negotiation among agents and the exchange of the data have been evaluated using patients' data collected in the neonatal intensive care unit over five years within the frame of a routine TDM program in CHUV (the Lausanne University Hospital).

Apart from creating the datasets that can be found and reused with respect to the requirements of a study, the evaluation results demonstrate that the more time the system is in use, the closer it is to a "fair" state, i.e., when the difference between the number of records provided and the number of records obtained (by an agent) converges to zero. We believe that using our MAS, compared to using a single database, does not differ significantly from the user point of view. However, the advantage of using our system is that it offers an access to more data, in a shorter period of time and in a privacy-preserving way.

7.2 Future Work

Our work is a solid basis for extending the results presented in this thesis in the following directions. For the realization of the application of the blockchain technology in connected

health, our framework, developed for primary care and medical research, could be expanded and combined with semantic technologies and principles of building multi-agent systems. We envisage a seamless integration and the traceability of different healthcare data streams, carried out by intelligent agents. This would facilitate the development of mobile health and optimize the management of patients data. However, currently, there is no existing legislation that governs the use of blockchain technology. This could become an obstacle that would slow down the implementation of such a framework in a connected health scenario.

We could further improve the current state of the art of the privacy-preserving microdata aggregation and, in particular, enhance precision medicine. This could be achieved by developing an adaptive framework that defines parameters for data anonymization on the fly by using different characteristics of data and data sources. This would lead to the utility improvement of the anonymized data. However, this is quite challenging as it would also require constructing the background knowledge of the adversary in an adaptive manner, in order to minimize de-identification risks and to guarantee the desired privacy level. It is also very important to inform a large audience about the results of the studies concerning the assessment of the privacy risks and the benefits for the patient and his social environment when he shares the data for research purposes. Designing a framework that would enable a personalized privacy risk assessment based on the health status of the patient in a privacy-preserving manner could help the patient to define the appropriate access-control; this would also facilitate the management of his medical history.

In the framework of therapeutic drug monitoring and establishing the interoperability in a clinical environment, we envisage the use of genomic data to complement the existing tools for automated TDM and the data acquisition for developing new models for other drugs (candidates for TDM). Coupling the TDM software with a point-of-care system would require more focus on the data security when the system sends the non-anonymized data to the caregiver such as alerts in case of emergency.

EHealth is a powerful and emerging field built at the intersection of healthcare, information and communication technologies, business, and social science. Interdisciplinary research collaborations are very important in order to put into practice and fully benefit from eHealth. We believe that the current work will help advance intelligent privacy-preserving data mining of the patients' health information and help improve healthcare.

Bibliography

[ABB+18] Elli Androulaki, Artem Barger, Vita Bortnikov, Christian Cachin, Konstantinos Christidis, Angelo De Caro, David Enyeart, Christopher Ferris, Gennady Laventman, Yacov Manevich, et al. Hyperledger fabric: A distributed operating system for permissioned blockchains. *arXiv preprint arXiv:1801.10228*, 2018.

[ABCP15] Michel Abdalla, Florian Bourse, Angelo De Caro, and David Pointcheval. Simple functional encryption schemes for inner products. *IACR Cryptology ePrint Archive*, 2015:17, 2015.

[ABG+17] Manel Aouri, Catalina Barcelo, Monia Guidi, Margalida Rotger, Matthias Cavassini, Cédric Hizrel, Thierry Buclin, Laurent A Decosterd, Chantal Csajka, Swiss HIV Cohort Study, et al. Population pharmacokinetics and pharmacogenetics analysis of rilpivirine in hiv-1-infected individuals. *Antimicrobial agents and chemotherapy*, 61(1):e00899–16, 2017.

[ABT13] Sima Ajami and Tayyebe Bagheri-Tadi. Barriers for adopting electronic health records (ehrs) by physicians. *Acta Informatica Medica*, 21(2):129, 2013.

[AEVL16] Asaph Azaria, Ariel Ekblaw, Thiago Vieira, and Andrew Lippman. Medrec: Using blockchain for medical data access and permission management. In *Open and Big Data (OBD), International Conference on*, pages 25–30. IEEE, 2016.

[AKRKG13] Harald Aamot, Christian Dominik Kohl, Daniela Richter, and Petra Knaup-Gregori. Pseudonymization of patient identifiers for translational research. *BMC medical informatics and decision making*, 13(1):75, January 2013.

[AMM17] Basma Al-Metwali and Hussain Mulla. Personalised dosing of medicines for children. *Journal of Pharmacy and Pharmacology*, 2017.

[B+16] Katherine Bourzac et al. Power to the patients, 2016.

[BA05] Roberto J Bayardo and Rakesh Agrawal. Data privacy through optimal k-anonymization. In *Data Engineering, 2005. ICDE 2005. Proceedings. 21st International Conference on*, pages 217–228. IEEE, 2005.

[BB10] Albert Boonstra and Manda Broekhuis. Barriers to the acceptance of electronic medical records by physicians from systematic review to taxonomy and interventions. *BMC health services research*, 10(1):231, 2010.

[BBH06] Dan Boneh, Xavier Boyen, and Shai Halevi. Chosen ciphertext secure public key threshold encryption without random oracles. In *Cryptographers Track at the RSA Conference*, pages 226–243. Springer, 2006.

[BCKL08] Mira Belenkiy, Melissa Chase, Markulf Kohlweiss, and Anna Lysyanskaya. P-signatures and noninteractive anonymous credentials. In *Theory of Cryptography Conference*, pages 356–374. Springer, 2008.

[BDCOP04] Dan Boneh, Giovanni Di Crescenzo, Rafail Ostrovsky, and Giuseppe Persiano. Public key encryption with keyword search. In *Eurocrypt*, volume 3027, pages 506–522. Springer, 2004.

[BDFMS02] Ruth Brand, J Domingo-Ferrer, and JM Mateo-Sanz. Reference data sets to test and compare sdc methods for protection of numerical microdata. *European Project IST-2000-25069 CASC*, 2002.

[Bec08] Georg Becker. Merkle signature schemes, merkle trees and their cryptanalysis. *Ruhr-University Bochum, Tech. Rep*, 2008.

[BHY15] Johan Gustav Bellika, Torje Starbo Henriksen, and Kassaye Yitbarek Yigzaw. The snow system: a decentralized medical data processing system. *Data Mining in Clinical Medicine*, pages 109–122, 2015.

[BI16] Paul Beninger and Michael A Ibara. Pharmacovigilance and biomedical informatics: a model for future development. *Clinical Therapeutics*, 2016.

[Bis03] Matt Bishop. *Computer security: art and science*. Addison-Wesley Professional, 2003.

[BK11] G. R. Blakley and Gregory Kabatiansky. *Secret Sharing Schemes*. Springer US, 2011.

[BKKJ+17] Maria Borge, Eleftherios Kokoris-Kogias, Philipp Jovanovic, Linus Gasser, Nicolas Gailly, and Bryan Ford. Proof-of-personhood: Redemocratizing permissionless cryptocurrencies. In *Security and Privacy Workshops (EuroS&PW), 2017 IEEE European Symposium on*, pages 23–26. IEEE, 2017.

[BLJ08] Elisa Bertino, Dan Lin, and Wei Jiang. A survey of quantification of privacy preserving data mining algorithms. In *Privacy-preserving data mining*, pages 183–205. Springer, 2008.

[BLLW11] Muzammil M Baig, Jiuyong Li, Jixue Liu, and Hua Wang. Cloning for privacy protection in multiple independent data publications. pages 885–894, 2011.

[BMC+15] Joseph Bonneau, Andrew Miller, Jeremy Clark, Arvind Narayanan, Joshua A Kroll, and Edward W Felten. Sok: Research perspectives and challenges for bitcoin and cryptocurrencies. In *Security and Privacy (SP), 2015 IEEE Symposium on*, pages 104–121. IEEE, 2015.

[BMW03] Mihir Bellare, Daniele Micciancio, and Bogdan Warinschi. Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 614–629. Springer, 2003.

[BS08] Justin Brickell and Vitaly Shmatikov. The cost of privacy: destruction of data-mining utility in anonymized data publishing. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 70–78. ACM, 2008.

[BS14] Emmanuel Benoist and Jan Sliwa. How to Collect Consent for an Anonymous Medical Database. *HEALTHINF*, 2014.

[BSB94] A. J. Boeckmann, L. B. Sheiner, and S. L. Beal. Nonmem users guide. *San Francisco: University of California San Francisco*, 1994.

[BSBL06] Ji-Won Byun, Yonglak Sohn, Elisa Bertino, and Ninghui Li. Secure anonymization for incremental datasets. *Secure Data Management*, 6:48–63, 2006.

[But14] Vitalik Buterin. Ethereum: A next-generation smart contract and decentralized application platform. *URL https://github. com/ethereum/wiki/wiki/% 5BEnglish% 5D-White-Paper*, 2014.

[CABC+12] Olivier Capitain, Andreaa Asevoaia, Michele Boisdron-Celle, Anne-Lise Poirier, Alain Morel, and Erick Gamelin. Individual fluorouracil dose adjustment in folfox based on pharmacokinetic follow-up compared with conventional body-area-surface dosing: a phase ii, proof-of-concept study. *Clinical colorectal cancer*, 11(4):263–267, 2012.

[Cac16] Christian Cachin. Architecture of the hyperledger blockchain fabric. 2016.

[CGH04] Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited. *Journal of the ACM (JACM)*, 51(4):557–594, 2004.

[Cha84] David Chaum. Blind signature system. In *Advances in cryptology*, pages 153–153. Springer, 1984.

[CHL+12] Neil Calman, Diane Hauser, Joseph Lurio, Winfred Y Wu, and Michelle Pichardo. Strengthening public health and primary care collaboration through electronic health records. *American journal of public health*, 102(11):e13–e18, 2012.

[CJ05] Christopher Clifton and Wei Jiang. CERIAS Tech Report 2005-134 Information Assurance and Security Privacy-Preserving Distributed k -Anonymity. 2005.

[CL02] Miguel Castro and Barbara Liskov. Practical byzantine fault tolerance and proactive recovery. *ACM Transactions on Computer Systems (TOCS)*, 20(4):398–461, 2002.

[CL17] Jan Camenisch and Anja Lehmann. Privacy for distributed databases via (un) linkable pseudonyms. *IACR Cryptology ePrint Archive*, 2017:22, 2017.

[CM03] R.M. Califf and L.H. Muhlbaier. Health insurance portability and accountability act (hipaa): Must there be a trade-off between privacy and quality of health care, or can we advance both? *Circulation*, 108(8):915–918, 2003.

[CMGP10] Víctor H Castillo, Ana I Martínez-García, and JRG Pulido. A knowledge-based taxonomy of critical factors for adopting electronic health record systems by physicians: a systematic literature review. *BMC medical informatics and decision making*, 10(1):60, 2010.

[CMS+17] Davide Calvaresi, Mauro Marinoni, Arnon Sturm, Michael Schumacher, and Giorgio Buttazzo. The challenge of real-time multi-agent systems for enabling iot and cps. *in proceedings of IEEE/WIC/ACM International Conference on Web Intelligence (WI'17)*, August 2017.

[COZ99] P. Ciancarini, A. Omicini, and F. Zambonelli. Multiagent system engineering: The coordination viewpoint. In *International Workshop on Agent Theories, Architectures, and Languages*, 1999.

[CSLL10] Graham Cormode, Divesh Srivastava, Ninghui Li, and Tiancheng Li. Minimizing minimality and maximizing utility: analyzing method-based attacks on anonymized data. *Proceedings of the VLDB Endowment*, 3(1-2):1045–1056, 2010.

[CSM+15] Giulia Cappi, Fabio M Spiga, Yessica Moncada, Anna Ferretti, Michael Beyeler, Marco Bianchessi, Laurent Decosterd, Thierry Buclin, and Carlotta Guiducci. Label-free detection of tobramycin in serum by transmission-localized surface plasmon resonance. *Analytical chemistry*, 87(10):5278–5285, 2015.

[CV17] Christian Cachin and Marko Vukolić. Blockchains consensus protocols in the wild. *arXiv preprint arXiv:1707.01873*, 2017.

[DBS+17] Alevtina Dubovitskaya, Thierry Buclin, Michael Schumacher, Karl Aberer, and Yann Thoma. Tucuxi: An intelligent system for personalized medicine from individualization of treatments to research databases and back. In *Proceedings of the 8th ACM International Conference on Bioinformatics, Computational Biology,and Health Informatics*, ACM-BCB '17, pages 223–232, New York, NY, USA, 2017. ACM.

[DDFS02] Ramesh A Dandekar, Josep Domingo-Ferrer, and Francesc Sebé. Lhs-based hybrid microdata vs rank swapping and microaggregation for numeric microdata protection. *Inference Control in Statistical Databases*, 2316:153–162, 2002.

[DFGN10] Josep Domingo-Ferrer and Úrsula González-Nicolás. Hybrid microdata using microaggregation. *Information Sciences*, 180(15):2834–2844, 2010.

[DFMS02] Josep Domingo-Ferrer and Josep Maria Mateo-Sanz. Practical data-oriented microaggregation for statistical disclosure control. *IEEE Transactions on Knowledge and data Engineering*, 14(1):189–201, 2002.

[DH76] Whitfield Diffie and Martin Hellman. New directions in cryptography. *IEEE transactions on Information Theory*, 22(6):644–654, 1976.

[DLLK+11] S De Lusignan, S-T Liaw, P Krause, V Curcin, M Tristan Vicente, G Michalakidis, L Agreus, P Leysen, N Shaw, and K Mendis. Key concepts to assess the readiness of data for international research: Data quality, lineage and provenance, extraction and processing errors, traceability, and curation. *Yearbook of medical informatics*, 20(01):112–120, 2011.

[dlTLAPV13] Albert Brugués de la Torre, Magi Lluch-Ariet, and Josep Pegueroles-Vallés. Security analysis of a protocol based on multiagents systems for clinical data exchange. In *Complex, Intelligent, and Software Intensive Systems (CISIS), 2013 Seventh International Conference on*, pages 305–311. IEEE, 2013.

[DMDMRF08] Filip De Meyer, Georges De Moor, and L Reed-Fourquet. Privacy protection through pseudonymisation in ehealth. In *HIT@ HealthCare 2008 joint event: 25th MIC congress; 3rd International congress Sixi; Special ISV-NVKVV event; 8th Belgian eHealth symposium*, volume 141, pages 111–118. IOS Press, 2008.

[DMNS06] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *TCC*, volume 3876, pages 265–284. Springer, 2006.

[DMRS+15] Yves-Alexandre De Montjoye, Laura Radaelli, Vivek Kumar Singh, et al. Unique in the shopping mall: On the reidentifiability of credit card metadata. *Science*, 347(6221):536–539, 2015.

[DMS04] Roger Dingledine, Nick Mathewson, and Paul Syverson. Tor: The second-generation onion router. 2004.

[DNO98] Enrico Denti, Antonio Natali, and Andrea Omicini. On the expressive power of a language for programming coordination media. In *Proceedings of the 1998 ACM symposium on Applied Computing*, pages 169–177. ACM, 1998.

[DUB+16] Alevtina Dubovitskaya, Visara Urovi, Imanol Barba, Karl Aberer, and Michael Ignaz Schumacher. A multiagent system for dynamic data aggregation in medical research. *BioMed Research International*, 2016, 2016.

[DUV+14] Alevtina Dubovitskaya, Visara Urovi, Matteo Vasirani, Karl Aberer, Aline Fuchs, Thierry Buclin, Yann Thoma, and Michael Schumacher. Privacy preserving interoperability for personalized medicine. Swiss Medical Informatics, September 2014.

[DUV+15] Alevtina Dubovitskaya, Visara Urovi, Matteo Vasirani, Karl Aberer, and Michael I Schumacher. A cloud-based ehealth architecture for privacy preserving data integration. In *IFIP International Information Security Conference*, pages 585–598. Springer, 2015.

[DXR+17a] Alevtina Dubovitskaya, Zhigang Xu, Samuel Ryu, Michael Schumacher, and Fusheng Wang. Blockchain dans la esanté: perspectives et une application pour le traitement quotidien. *Swiss Medical Informatics*, 33, 2017.

[DXR+17b] Alevtina Dubovitskaya, Zhigang Xu, Samuel Ryu, Michael Schumacher, and Fusheng Wang. How blockchain could empower ehealth: An application for radiation oncology. In *VLDB Workshop on Data Management and Analytics for Medicine and Healthcare*, pages 3–6. Springer, 2017.

[DXR+17c] Alevtina Dubovitskaya, Zhigang Xu, Samuel Ryu, Michael Schumacher, and Fusheng Wang. Secure and trustable electronic medical records sharing using blockchain. In *AMIA Annual Symposium Proceedings*, volume 2017, page 650. American Medical Informatics Association, 2017.

[EERM15] Khaled El Emam, Sam Rodgers, and Bradley Malin. Anonymising and sharing individual patient data. *bmj*, 350:h1139, 2015.

[EFGK03] Patrick Th Eugster, Pascal A Felber, Rachid Guerraoui, and Anne-Marie Kermarrec. The many faces of publish/subscribe. *ACM computing surveys (CSUR)*, 35(2):114–131, 2003.

[EGSVR16] Ittay Eyal, Adem Efe Gencer, Emin Gün Sirer, and Robbert Van Renesse. Bitcoin-ng: A scalable blockchain protocol. In *NSDI*, pages 45–59, 2016.

[EII+10] Bernice S Elger, Jimison Iavindrasana, Luigi Lo Iacono, Henning Müller, Nicolas Roduit, Paul Summers, and Jessica Wright. Strategies for health data exchange for secondary, cross-institutional clinical research. *Computer methods and programs in biomedicine*, 99(3):230–251, 2010.

[Eur16] Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation). *Official Journal of the European Union*, L119/59, May 2016.

[FCT+12] A. Fuchs, C. Csajka, Y. Thoma, T. Buclin, and N. Widmer. Benchmarking therapeutic drug monitoring software: A review of available computer tools. *Clinical Pharmacokinetics*, 52(1):9–22, 2012.

[FCT+13] Aline Fuchs, Chantal Csajka, Yann Thoma, Thierry Buclin, and Nicolas Widmer. Benchmarking therapeutic drug monitoring software: a review of available computer tools. *Clinical pharmacokinetics*, 52(1):9–22, 2013.

[FES+17] David Froelicher, Patricia Egger, João Sá Sousa, Jean Louis Raisaro, Zhicong Huang, Christian Mouchet, Bryan Ford, and Jean-Pierre Hubaux. Unlynx: a decentralized system for privacy-conscious data sharing. *Proceedings on Privacy Enhancing Technologies*, 2017(4):232–250, 2017.

[FGG+14] Aline Fuchs, Monia Guidi, Eric Giannoni, Dominique Werner, Thierry Buclin, Nicolas Widmer, and Chantal Csajka. Population pharmacokinetic study of gentamicin in a large cohort of premature and term neonates. *British journal of clinical pharmacology*, 78(5):1090–1101, 2014.

[FLJ+14] Matthew Fredrikson, Eric Lantz, Somesh Jha, Simon Lin, David Page, and Thomas Ristenpart. Privacy in pharmacogenetics: An end-to-end case study of personalized warfarin dosing. In *USENIX Security Symposium*, pages 17–32, 2014.

[FWY05] Benjamin CM Fung, Ke Wang, and Philip S Yu. Top-down specialization for information and privacy preservation. In *Data Engineering, 2005. ICDE 2005. Proceedings. 21st International Conference on*, pages 205–216. IEEE, 2005.

[GD09] Patrick Gallagher and Cita Furlani Director. Fips pub 186-3 federal information processing standards publication digital signature standard (dss), 2009.

[GDLS14] Aris Gkoulalas-Divanis, Grigorios Loukides, and Jimeng Sun. Publishing data from electronic health records while preserving privacy: A survey of algorithms. *Journal of biomedical informatics*, 50:4–19, 2014.

[Gel85] David Gelernter. Generative communication in linda. *ACM Trans. Program. Lang. Syst.*, 7(1):80–112, January 1985.

[GHM+17] Yossi Gilad, Rotem Hemo, Silvio Micali, Georgios Vlachos, and Nickolai Zeldovich. Algorand: Scaling byzantine agreements for cryptocurrencies. SOSP, 2017.

[GKS08] Srivatsava Ranjit Ganta, Shiva Prasad Kasiviswanathan, and Adam Smith. Composition attacks and auxiliary information in data privacy. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, pages 265–273, New York, NY, USA, 2008. ACM.

[GKW+16] Arthur Gervais, Ghassan O Karame, Karl Wüst, Vasileios Glykantzis, Hubert Ritzdorf, and Srdjan Capkun. On the security and performance of proof of work blockchains. pages 3–16, 2016.

[GL13] Aris Gkoulalas-Divanis and Grigorios Loukides. *Anonymization of Electronic Medical Records to Support Clinical Analysis*. Springer Briefs in Electrical and Computer Engineering. Springer, 2013.

[GMN+16] Paul Grubbs, Richard McPherson, Muhammad Naveed, Thomas Ristenpart, and Vitaly Shmatikov. Breaking web applications built on top of encrypted data. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 1353–1364. ACM, 2016.

[Gol09] Oded Goldreich. *Foundations of cryptography: volume 2, basic applications*. Cambridge university press, 2009.

[GPW+14] David H Goldstein, Rachel Phelan, Rosemary Wilson, Amanda Ross-White, Elizabeth G VanDenKerkhof, John P Penning, and Melanie Jaeger. Brief review: Adoption of electronic medical records to enhance acute pain management. *Canadian Journal of Anesthesia/Journal canadien d'anesthésie*, 61(2):164–179, 2014.

[GSAA04] Abhishek Gupta, Ozgur D Sahin, Divyakant Agrawal, and Amr El Abbadi. Meghdoot: content-based publish/subscribe over p2p networks. In *Proceedings of the 5th ACM/IFIP/USENIX international conference on Middleware*, pages 254–273. 2004.

[GWM+12] Verena Gotta, Nicolas Widmer, Michael Montemurro, Serge Leyvraz, Amina Haouala, Laurent A Decosterd, Chantal Csajka, and Thierry Buclin. Therapeutic drug monitoring of imatinib. *Clinical pharmacokinetics*, 51(3):187–201, 2012.

[HDFF+12] Anco Hundepool, Josep Domingo-Ferrer, Luisa Franconi, Sarah Giessing, Eric Schulte Nordholt, Keith Spicer, and Peter-Paul de Wolf. *Microdata*, pages 23–130. John Wiley & Sons, Ltd, 2012.

[HMBW13] Gregory William Hruby, James McKiernan, Suzanne Bakken, and Chunhua Weng. A centralized research data repository enhances retrospective outcomes research capacity: a case report. *Journal of the American Medical Informatics Association*, 20(3):563–567, 2013.

[HMRB15] Meskerem Asfaw Hailemichael, Luis Marco-Ruiz, and Johan Gustav Bellika. Privacy-preserving statistical query and processing on distributed openehr data. *Studies in health technology and informatics*, 210:766–770, 2015.

[HN09] Yeye He and Jeffrey F. Naughton. Anonymization of set-valued data via top-down, local generalization. *Proc. VLDB Endow.*, 2(1):934–945, August 2009.

[Hod16] Richard Hodson. Precision medicine. *Nature*, 537(7619):S49, 2016.

[HS00] Martin Hirt and Kazue Sako. Efficient receipt-free voting based on homomorphic encryption. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 539–556. Springer, 2000.

[HT93] Vassos Hadzilacos and Sam Toueg. Distributed systems. *Fault-Tolerant Broadcasts and Related Problems*, pages 97–145, 1993.

[HX94] Lein Harn and Y Xu. Design of generalised elgamal type digital signature schemes based on discrete logarithm. *Electronics letters*, 30(24):2025–2026, 1994.

[IAP09] Luan Ibraimi, Muhammad Asim, and Milan Petković. Secure management of personal health records by applying attribute-based encryption. In *Wearable Micro and Nano Technologies for Personalized Health (pHealth), 2009 6th International Workshop on*, pages 71–74. IEEE, 2009.

[Jel91] Roger W Jelliffe. The usc* pack pc programs for population pharmacokinetic modeling, modeling of large kinetic/dynamic systems, and adaptive control of drug dosage regimens. In *Proceedings of the Annual Symposium on Computer Application in Medical Care*, page 922. American Medical Informatics Association, 1991.

[JH03] Denis Jouvin and Salima Hassas. Dynamic multi-agent architecture using conversational role delegation. In *Agent-Oriented Software Engineering IV*, pages 185–200. Springer, 2003.

[JKT+15] Jeffrey Jenkins, Jarad Kopf, Binh Q Tran, Christopher Frenchi, and Harold Szu. Bio-mining for biomarkers with a multi-resolution block chain. In *SPIE Sensing Technology+ Applications*, pages 94960N–94960N. International Society for Optics and Photonics, 2015.

[KKOM17] Tsung-Ting Kuo, Hyeon-Eui Kim, and Lucila Ohno-Machado. Blockchain distributed ledger technologies for biomedical and health care applications. *Journal of the American Medical Informatics Association*, 2017.

[KL09] Ju-Seop Kang and Min-Ho Lee. Overview of therapeutic drug monitoring. *The Korean journal of internal medicine*, 24(1):1–10, 2009.

[KP16] Aggelos Kiayias and Giorgos Panagiotakos. On trees, chains and fast transactions in the blockchain. *IACR Cryptology ePrint Archive*, 2016:545, 2016.

[KPE+12] Florian Kohlmayer, Fabian Prasser, Claudia Eckert, Alfons Kemper, and Klaus A Kuhn. Highly efficient optimal k-anonymity for biomedical datasets. In *Computer-Based Medical Systems (CBMS), 2012 25th International Symposium on*, pages 1–6. IEEE, 2012.

[KRDO17] Aggelos Kiayias, Alexander Russell, Bernardo David, and Roman Oliynykov. Ouroboros: A provably secure proof-of-stake blockchain protocol. In *Annual International Cryptology Conference*, pages 357–388. Springer, 2017.

[Lam98] Leslie Lamport. The part-time parliament. *ACM Transactions on Computer Systems (TOCS)*, 16(2):133–169, 1998.

[LBC16] James J Lee, Jan H Beumer, and Edward Chu. Therapeutic drug monitoring of 5-fluorouracil. *Cancer chemotherapy and pharmacology*, 78(3):447–464, 2016.

[LCHL11] Zhuo-Rong Li, En-Chi Chang, Kuo-Hsuan Huang, and Feipei Lai. A secure electronic medical record sharing mechanism in the cloud computing platform. In *Consumer Electronics (ISCE), 2011 IEEE 15th International Symposium on*, pages 98–103. IEEE, 2011.

[LDR05] Kristen LeFevre, David J DeWitt, and Raghu Ramakrishnan. Incognito: Efficient full-domain k-anonymity. In *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, pages 49–60. ACM, 2005.

[LDR06] Kristen LeFevre, David J DeWitt, and Raghu Ramakrishnan. Mondrian multidimensional k-anonymity. In *Data Engineering, 2006. ICDE'06. Proceedings of the 22nd International Conference on*, pages 25–25. IEEE, 2006.

[LFS+10] Evette J Ludman, Stephanie M Fullerton, Leslie Spangler, Susan Brown Trinidad, Monica M Fujii, Gail P Jarvik, Eric B Larson, and Wylie Burke. Glad you asked: participants' opinions of re-consent for dbgap data submission. *Journal of Empirical Research on Human Research Ethics*, 5(3):9–16, 2010.

[LHBC13] Ahmed Lounis, Abdelkrim Hadjidj, Abdelmadjid Bouabdallah, and Yacine Challal. Secure medical architecture on the cloud using wireless sensor networks for emergency management. In *Broadband and Wireless Computing, Communication and Applications (BWCCA), 2013 Eighth International Conference on*, pages 248–252. IEEE, 2013.

[LL09] Tiancheng Li and Ninghui Li. On the tradeoff between privacy and utility in data publishing. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 517–526. ACM, 2009.

[LLV07] Ninghui Li, Tiancheng Li, and Suresh Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and l-diversity. In *Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on*, pages 106–115. IEEE, 2007.

[Lo 07] Luigi Lo Iacono. Multi-centric universal pseudonymisation for secondary use of the EHR. *Studies in health technology and informatics*, 126:239–47, January 2007.

[LOW08] Jianzhong Li, Beng Chin Ooi, and Weiping Wang. Anonymizing streaming data for privacy protection. In *Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on*, pages 1367–1369. IEEE, 2008.

[LSP82] Leslie Lamport, Robert Shostak, and Marshall Pease. The byzantine generals problem. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 4(3):382–401, 1982.

[LYRL10] Ming Li, Shucheng Yu, Kui Ren, and Wenjing Lou. Securing personal health records in cloud computing: Patient-centric and fine-grained data access control in multi-owner settings. 10:89–106, 2010.

[LYZ+13] Ming Li, Shucheng Yu, Yao Zheng, Kui Ren, and Wenjing Lou. Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption. *IEEE transactions on parallel and distributed systems*, 24(1):131–143, 2013.

[Maz15] David Mazieres. The stellar consensus protocol: A federated model for internet-level consensus. *Stellar Development Foundation*, 2015.

[Mer80] Ralph C Merkle. Protocols for public key cryptosystems. In *Security and Privacy, 1980 IEEE Symposium on*, pages 122–122. IEEE, 1980.

[MGKV06] Ashwin Machanavajjhala, Johannes Gehrke, Daniel Kifer, and Muthuramakrishnan Venkitasubramaniam. l-diversity: Privacy beyond k-anonymity. In *Data Engineering, 2006. ICDE'06. Proceedings of the 22nd International Conference on*, pages 24–24. IEEE, 2006.

[MOO+14] Ciara Moore, Maire O'Neill, Elizabeth O'Sullivan, Yarkin Doroz, and Berk Sunar. Practical homomorphic encryption: A survey. In *Circuits and Systems (ISCAS), 2014 IEEE International Symposium on*, pages 2792–2795. IEEE, 2014.

[MRV99] Silvio Micali, Michael Rabin, and Salil Vadhan. Verifiable random functions. In *Foundations of Computer Science, 1999. 40th Annual Symposium on*, pages 120–130. IEEE, 1999.

[MT95] Y. Moses and M. Tennenholtz. Artificial social systems. *Computers and Artificial Intelligence*, 1995.

[MVOV96] Alfred J Menezes, Paul C Van Oorschot, and Scott A Vanstone. *Handbook of applied cryptography*. CRC press, 1996.

[MW14] J D Momper and J A Wagner. Therapeutic drug monitoring as a component of personalized medicine: Applications in pediatric drug development. *Clinical Pharmacology & Therapeutics*, 95(2):138–140, 2014.

[MWM+10] Shawn N Murphy, Griffin Weber, Michael Mendis, Vivian Gainer, Henry C Chueh, Susanne Churchill, and Isaac Kohane. Serving the enterprise and beyond with informatics for integrating biology and the bedside (i2b2). *Journal of the American Medical Informatics Association*, 17(2):124–130, 2010.

[NAC07] Mehmet Ercan Nergiz, Maurizio Atzori, and Chris Clifton. Hiding the presence of individuals from shared databases. In *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, pages 665–676. ACM, 2007.

[Nak08] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system, 2008.

[NAT14] Guillermo Navarro-Arribas and Vicenç Torra. Rank swapping for stream data. In *Modeling Decisions for Artificial Intelligence*, pages 217–226. Springer, 2014.

[NCW+17] Fahima Nekka, Chantal Csajka, Mélanie Wilbaux, Sachin Sanduja, Jun Li, and Marc Pfister. Pharmacometrics-based decision tools facilitate mhealth implementation. *Expert review of clinical pharmacology*, 10(1):39–46, 2017.

[NDNTA14] Quoc Viet Hung Nguyen, Son Thanh Do, Tam Nguyen Thanh, and Karl Aberer. Privacy-preserving schema reuse. In *Database Systems For Advanced Applications, Dasfaa 2014, Pt Ii*, volume 8422, pages 234–250. Springer-Verlag Berlin, 2014.

[NH11] Thomas Neubauer and Johannes Heurix. A methodology for the pseudonymization of medical data. *International journal of medical informatics*, 80(3):190–204, 2011.

[NLL07] Rita Noumeir, Alain Lemay, and Jean-Marc Lina. Pseudonymization of radiology data for research purposes. *Journal of digital imaging*, 20(3):284–295, 2007.

[NLM+13] Hung Quoc Viet Nguyen, Xuan Hoai Luong, Zoltán Miklós, Tho Quan Thanh, and Karl Aberer. An mas negotiation support tool for schema matching. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, pages 1391–1392. International Foundation for Autonomous Agents and Multiagent Systems, 2013.

[NS08] Arvind Narayanan and Vitaly Shmatikov. Robust de-anonymization of large sparse datasets. In *Proceedings of the 2008 IEEE Symposium on Security and Privacy*, SP '08, pages 111–125, Washington, DC, USA, 2008. IEEE Computer Society.

[NVP10] Elena Nardini, Mirko Viroli, and Emanuele Panzavolta. Coordination in open and dynamic environments with tucson semantic tuple centres. In *Proceedings of the 2010 ACM Symposium on Applied Computing*, pages 2037–2044. ACM, 2010.

[O+16] World Health Organization et al. From innovation to implementation: ehealth in the who european region 2016. *Copenhagen: WHO Regional Office for Europe*, 2016.

[Oku14] Krzysztof Okupski. Bitcoin developer reference. *Eindhoven*, 2014.

[OL88] Brian M Oki and Barbara H Liskov. Viewstamped replication: A new primary copy method to support highly-available distributed systems. In *Proceedings of the seventh annual ACM Symposium on Principles of distributed computing*, pages 8–17. ACM, 1988.

[OM13] Karl J O'Dwyer and David Malone. Bitcoin mining and its energy footprint. In *Irish Signals & Systems Conference 2014 and 2014 China-Ireland International Conference on Information and Communications Technologies (ISSC 2014/CIICT 2014). 25th IET*, pages 280–285. IET, 2013.

[OO14] Diego Ongaro and John K Ousterhout. In search of an understandable consensus algorithm. In *USENIX Annual Technical Conference*, pages 305–319, 2014.

[OZ99] Andrea Omicini and Franco Zambonelli. Coordination for internet application development. *Autonomous Agents and Multi-agent systems*, 2(3):251–269, 1999.

[Per09] Colin Percival. Stronger key derivation via sequential memory-hard functions. *Self-published*, pages 1–16, 2009.

[PLGDS13] Giorgos Poulis, Grigorios Loukides, Aris Gkoulalas-Divanis, and Spiros Skiadopoulos. Anonymizing data with relational and transaction attributes. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 353–369. Springer, 2013.

[PM92] Johannes H Proost and Dirk KF Meijer. Mw/pharm, an integrated software package for drug dosage regimen calculation and therapeutic drug monitoring. *Computers in biology and medicine*, 22(3):155–163, 1992.

[PPT17] Angelo Massimo Perillo, Giuseppe Persiano, and Alberto Trombetta. Secure queries on encrypted multi-writer tables. In *Security and Privacy (EuroS&P), 2017 IEEE European Symposium on*, pages 127–141. IEEE, 2017.

[PRDS05] K Pommerening, M Reng, P Debold, and S Semler. Pseudonymization in medical research-the generic data protection concept of the tmf. *GMS Medizinische Informatik, Biometrie und Epidemiologie*, 1(3):2005–1, 2005.

[PS17] Rafael Pass and Elaine Shi. Hybrid consensus: Efficient consensus in the permissionless model. In *LIPIcs-Leibniz International Proceedings in Informatics*, volume 91. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2017.

[PXW+07] Jian Pei, Jian Xu, Zhibin Wang, Wei Wang, and Ke Wang. Maintaining k-anonymity against incremental updates. In *Scientific and Statistical Database Management, 2007. SSBDM'07. 19th International Conference on*, pages 5–5. IEEE, 2007.

[PZ13] Raluca Ada Popa and Nickolai Zeldovich. Multi-key searchable encryption. *Cryptology ePrint Archive, Report 2013/508*, 2013.

[RHJ04] Sarvapali D Ramchurn, Dong Huynh, and Nicholas R Jennings. Trust in multi-agent systems. *The Knowledge Engineering Review*, 19(1):1–25, 2004.

[RN03] Stuart J Russell and Peter Norving. Norvig. *Artificial Intelligence: A Modern Approach*, pages 111–114, 2003.

[RSH07] Vibhor Rastogi, Dan Suciu, and Sungho Hong. The boundary between privacy and utility in data publishing. In *Proceedings of the 33rd international conference on Very large data bases*, pages 531–542. VLDB Endowment, 2007.

[Sam01] Pierangela Samarati. Protecting respondents identities in microdata release. *IEEE transactions on Knowledge and Data Engineering*, 13(6):1010–1027, 2001.

[SBH+07] Charles Safran, Meryl Bloomrosen, W Edward Hammond, Steven Labkoff, Suzanne Markel-Fox, Paul C Tang, and Don E Detmer. Toward a national framework for the secondary use of health data: an american medical informatics association white paper. *Journal of the American Medical Informatics Association*, 14(1):1–9, 2007.

[SBY+14] Alena Simalatsar, Romain Bornet, Wenqi You, Yann Thoma, and Giovanni De Micheli. Safe implementation of embedded software for a portable device supporting drug administration. In *Bioinformatics and Bioengineering (BIBE), 2014 IEEE International Conference on*, pages 257–264. IEEE, 2014.

[SCDFSM14] Jordi Soria-Comas, Josep Domingo-Ferrer, David Sánchez, and Sergio Martínez. Enhancing data utility in differential privacy via microaggregation-based k-anonymity. *The VLDB Journal*, 23(5):771–794, 2014.

[Sch84] Ferdinand David Schoeman. *Philosophical dimensions of privacy: An anthology*. Cambridge University Press, 1984.

[Sch90] Fred B Schneider. Implementing fault-tolerant services using the state machine approach: A tutorial. *ACM Computing Surveys (CSUR)*, 22(4):299–319, 1990.

[SFK07] Y Sattarova Feruza and Tao-hoon Kim. It security review: Privacy, protection, access control, assurance and system security. *International journal of multimedia and ubiquitous engineering*, 2(2):17–31, 2007.

[SG98] Victor Shoup and Rosario Gennaro. Securing threshold cryptosystems against chosen ciphertext attack. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 1–16. Springer, 1998.

[SHB+12] Dean F Sittig, Brian L Hazlehurst, Jeffrey Brown, Shawn Murphy, Marc Rosenman, Peter Tarczy-Hornoch, and Adam B Wilcox. A survey of informatics platforms that enable distributed comparative effectiveness research using multi-institutional heterogeneous clinical data. *Medical care*, 50(Suppl):S49, 2012.

[SJAA+14] Marcos A Simplicio Jr, Leonardo C Almeida, Ewerton R Andrade, Paulo CF dos Santos, and Paulo SLM Barreto. The lyra2 reference guide. Technical report, version 2.3. 2. Technical report, 2014.

[SKC+07] Steven R Simon, Rainu Kaushal, Paul D Cleary, Chelsea A Jenter, Lynn A Volk, Eric G Poon, E John Orav, Helen G Lo, Deborah H Williams, and David W Bates. Correlates of electronic health record adoption in office practices: a statewide survey. *Journal of the American Medical Informatics Association*, 14(1):110–117, 2007.

[SMBMS13] Agusti Solanas, Antoni Martinez-Balleste, and Josep M Mateo-Sanz. Distributed architecture with double-phase microaggregation for the private sharing of biomedical data in mobile health. *IEEE Transactions on Information Forensics and Security*, 8(6):901–910, 2013.

[SRLH05] Margaret A Stone, Sarah A Redsell, Jennifer T Ling, and Alastair D Hay. Sharing patient data: competing demands of privacy, trust and research in primary care. *Br J Gen Pract*, 55(519):783–789, 2005.

[SS16] Murat Sariyar and Irene Schlunder. Reconsidering anonymization-related concepts and the term identification against the backdrop of the european legal framework. *Biopreservation and biobanking*, 2016.

[STV+16] Ewa Syta, Iulia Tamas, Dylan Visher, David Isaac Wolinsky, Philipp Jovanovic, Linus Gasser, Nicolas Gailly, Ismail Khoffi, and Bryan Ford. Keeping authorities "honest or bust" with decentralized witness cosigning. In *Security and Privacy (SP), 2016 IEEE Symposium on*, pages 526–545. Ieee, 2016.

[Swa15] Tim Swanson. Consensus-as-a-service: a brief report on the emergence of permissioned, distributed ledger systems, 2015.

[Swe02] Latanya Sweeney. K-anonymity: A model for protecting privacy. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, 10(5):557–570, October 2002.

[SWP00] Dawn Xiaoding Song, David Wagner, and Adrian Perrig. Practical techniques for searches on encrypted data. In *Security and Privacy, 2000. S&P 2000. Proceedings. 2000 IEEE Symposium on*, pages 44–55. IEEE, 2000.

[SYB14] David Schwartz, Noah Youngs, and Arthur Britto. The ripple protocol consensus algorithm. *Ripple Labs Inc White Paper*, 5, 2014.

[SZ15] Yonatan Sompolinsky and Aviv Zohar. Secure high-rate transaction processing in bitcoin. In *International Conference on Financial Cryptography and Data Security*, pages 507–527. Springer, 2015.

[UOB+12] Visara Urovi, Alex C Olivieri, Stefano Bromuri, Nicoletta Fornara, and Michael I Schumacher. An agent coordination framework for ihe based cross-community health record exchange. In *VII Workshop on Agents Applied in Health Care, A2HC 2012*, page 29, 2012.

[UOdlT+14] Visara Urovi, Alex C Olivieri, Albert Brugués de la Torre, Stefano Bromuri, Nicoletta Fornara, and Michael Schumacher. Secure p2p cross-community health record exchange in ihe compatible systems. *International Journal on Artificial Intelligence Tools*, 23(01):1440006, 2014.

[VBC+08] Marieke Verschuuren, Gérard Badeyan, Javier Carnicero, Mika Gissler, Renzo Pace Asciak, Luule Sakkeus, Magnus Stenbeck, Walter Deville, Work Group on Confidentiality, Data Protection of the Network of Competent Authorities of the Health Information, and Knowledge Strand of the EU Public Health Programme 2003-08. The european data protection legislation and its consequences for public health monitoring: a plea for action. *The European Journal of Public Health*, 18(6):550–551, 2008.

[Vuk15] Marko Vukolić. The quest for scalable blockchain fabric: Proof-of-work vs. bft replication. In *International Workshop on Open Problems in Network Security*, pages 112–125. Springer, 2015.

[Wal02] Gregory J Walters. *Human rights in an information age: A philosophical analysis*. University of Toronto Press, 2002.

[WD11] Daniel FB Wright and Stephen B Duffull. Development of a bayesian forecasting method for warfarin dose individualisation. *Pharmaceutical research*, 28(5):1100–1111, 2011.

[WF06] Ke Wang and Benjamin Fung. Anonymizing sequential releases. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 414–423. ACM, 2006.

[WHSW13] Nicole G Weiskopf, George Hripcsak, Sushmita Swaminathan, and Chunhua Weng. Defining and measuring completeness of electronic health records for secondary use. *Journal of biomedical informatics*, 46(5):830–836, 2013.

[WKS+15] Sebastian G Wicha, Martin G Kees, Alexander Solms, Iris K Minichmayr, Alexander Kratzer, and Charlotte Kloft. Tdmx: a novel web-based open-access sup-

port tool for optimising antimicrobial dosing regimens in clinical routine. *International journal of antimicrobial agents*, 45(4):442–444, 2015.

[Woo14] Gavin Wood. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum Project Yellow Paper*, 151:1–32, 2014.

[WVNL14] Fusheng Wang, Cristobal Vergara-Niedermayr, and Peiya Liu. Metadata based management and sharing of distributed biomedical data. *International journal of metadata, semantics and ontologies*, 9(1):42–57, 2014.

[XC14] Liangyu Xu and Armin B Cremers. A Decentralized Pseudonym Scheme for Cloud-based eHealth Systems. *HEALTHINF*, 2014.

[YWJ+16] Xiao Yue, Huiju Wang, Dawei Jin, Mingqiang Li, and Wei Jiang. Healthcare data gateways: found healthcare intelligence on blockchain with novel privacy risk control. *Journal of medical systems*, 40(10):218, 2016.

[YWRL10] Shucheng Yu, Cong Wang, Kui Ren, and Wenjing Lou. Achieving secure, scalable, and fine-grained data access control in cloud computing. In *Infocom, 2010 proceedings IEEE*, pages 1–9. Ieee, 2010.

[ZHP+09] Bin Zhou, Yi Han, Jian Pei, Bin Jiang, Yufei Tao, and Yan Jia. Continuous privacy preserving publishing of data streams. In *Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology*, pages 648–659. ACM, 2009.

[ZWC+16] Ennan Zhai, David Isaac Wolinsky, Ruichuan Chen, Ewa Syta, Chao Teng, and Bryan Ford. Anonrep: Towards tracking-resistant anonymous reputation. In *NSDI*, pages 583–596, 2016.

Alevtina Dubovitskaya

Address: Rue du Lac, 87, 1815,
Clarens (Switzerland)
Mobile: +41 78 881 53 56
Web: www.linkedin.com/in/aledbv
E-mail: alevtina.dubovitskaya@epfl.ch

RESEARCH INTERESTS

Digital Health, Blockchain, Privacy, Computer Security, Distributed Systems, Interoperability, Databases, Algorithms, Semantic Web

WORK EXPERIENCE

2013-p.t.	University of Applied Sciences Western Switzerland (HES-SO), *Research Assistant* • Work in the framework of an RTD project under Nano-Tera initiative: "Therapeutic drug monitoring for Personalized Medicine (ISyPeM2)"
2016 (Oct-Dec)	Stony Brook University (SBU), NY, USA, *Research Intern* • Work on application of the blockchain technology to support healthcare data management, in particular, EHR sharing between healthcare providers
2012-2013.	École Polytechnique Fédérale de Lausanne (EPFL), *Doctoral Assistant* Two Semester Projects in the Laboratory for Communications and Applications under the supervision of Dr. Kévin Huguenin and Prof. Jean-Pierre Hubaux
2011-2012	Pointlane LLC, Moscow, Russia, *Information security specialist* • Personal Data Protection, Information Security Audit, Information Security Products and Systems Implementation
2012 (Jan-June)	CRYPTO-PRO LLC, Moscow, Russia, *Intern* • Integration of Russian Standard for Encryption (GOST) into the Identity Management System

EDUCATION

2012-p.t.	École Polytechnique Fédérale de Lausanne (EPFL) *PhD student in Doctoral Program in Computer, Communication and Information Sciences* Research topic: Privacy-Preserving Interoperability for eHealth Systems
2014 (Oct-Dec)	Polytechnic University of Catalonia (UPC), *Intern* Research collaboration with Information Security Group

2007-2012	Moscow Engineering Physics Institute (National Research Nuclear University) *M.S.(eq) in Computer Science and Information Security* Research Work devoted to the Russian Standard for Encryption (GOST) Implementation in the Identity Management System
2010-2012	Moscow Engineering Physics Institute (National Research Nuclear University) *Translator in the Area of Professional Communication (English language)* Theory and Practice of Translation, Linguistics, Study of (the history, geography and culture of) UK and USA
2005-2007	Moscow Gymnasium № 1567 *Major: Biology, Mathematics, Chemistry* Winner of Mathematics, Biology and other Olympiads, graduated with excellence

TRAININGS

- Swiss blockchain summer school, EPFL, Lausanne, 2017
- NextStep PhD students Entrepreneurship program offered by Nano-Tera, Swiss federal research program, Lausanne, 2015-2016.
- Training modules Business Concept and Business Creation by Commission for Technology and Innovation (CTI) Entrepreneurship, Lausanne, 2015.
- The 16th European Agent Systems Summer School (EASSS), Crete, 2014

AWARDS

- Best Paper Awards for “Blockchain dans la eSanté: Perspectives et une Application pour le Traitement Quotidien” by Dubovitskaya, A et al. at Swiss eHealth Summit 2017,
- Best Poster Award for “Privacy Preserving Interoperability for Personalized Medicine”, by Dubovitskaya, A. et al at Swiss eHealth Summit 2014.
- Doctoral fellowship, EDIC program at EPFL, Lausanne, Switzerland (2012-2013)
- Russian Government Prize Fellowship (2011-2012)

PROFESSIONAL ACTIVITIES

- Workshop chair: Blockchain Technologies for Multi-Agent Systems (BCT4MAS)@WI’18
- Program committee member for the Workshop on Data Management and Analytics for Medicine and Healthcare (DMAH) in conj. with Conf. on Very Large Data Bases, VLDB’17,’18
- Reviewer for the IEEE journal Transactions on Information Forensics & Security, Computational and Structural Biotechnology Journal (Elsevier CSBJ), conferences: CLOSER’14, ICAART’15, ICTH’16, Netmob’17, workshops on Artificial Intelligence for Diabetes (AID)@ECAI’16, Real Time compliant Multi-Agent Systems (RTcMAS)@AAMAS’18

SKILLS

- Blockchain, computer and networks security, privacy, identity management, cryptography and PKI.
- Databases: SQL, parallel/distributed databases, concurrency control, interoperability.
- HL7 Standards.
- Programming: Java, Go, Python, UML. Experience in using Map-reduce/Hadoop.
- Languages: Russian (native); English (fluent); French (B2), German (A1), Italian (A1).
- Personal: Conscientious, reliable and goal-oriented person, eager to learn and self-improve every day.
- Communication skills: Able to work on own initiative and as a part of a team.
- First-class analytical, design and problem solving skills.
- Miscellaneous: Music: singing, piano (musical school with excellence), guitar, sport: Second-Class Junior Sportsman in artistic gymnastics, hiking, climbing, badminton.

PRESENTATIONS and MEDIA APPEARANCE

- "Privacy Preserving EMR Sharing Using Blockchain" – meetup, Lausanne, 2018
- "Blockchain & its potential applications in finance" – invited talk, Swissquote, 2018
- "Secure and Trustable EMR Sharing using Blockchain: Open Challenges and Lessons Learned" – Digital Health Connect, Sierre, 2018.
- "Blockchain: Organisatorische Aspekte im Bereich eHealth". Magglinger Rechtsinformatikseminar, 2017, Switzerland.
- "Blockchain dans la eSanté: Perspectives et une Application pour le Traitement Quotidien" – l'Université d'été de la eSanté, keynote, Castres, France, 2017.
- "La Blockchain pourrait révolutionner nos transactions", RTS 19:30.
- "Comment gérer les données oncologiques avec la blockchain", Conférence Technoark 2017 - Blockchain, au-delà du bitcoin, Sierre, Switzerland.
- "ISyPeM2 - de l'individualisation des posologies aux bases de données médicales", Journée e-Health 2016, Sierre, Switz., presented with Séverine Petitprez.
- "Integration of Russian Cryptographic Algorithms into the Identity Management System", student rump session, Women in Theory workshop, Princeton University, USA, 2012

PUBLICATIONS

(in English)

- Dubovitskaya, A., Calvaresi, D., Retaggi, D., Dragoni, A. F., and Schumacher, M. (2018). *Essais Clinique Multicentriques: Transparence et Contrôle de la Qualité Grâce à la Blockchain et les Systèmes Multi-Agents* (submitted).
- Dubovitskaya, A., Schumacher, M., Aberer, K. (2018). *Improving Utility of Incremental k-anonymous Datasets in Distributed Settings* (submitted).
- Calvaresi, D., Dubovitskaya, A., Retaggi, D., Dragoni, A. F., and Schumacher, M. (2018). *Trusted Registration, Negotiation, and Service Evaluation in Multi-Agent Systems throughout the Blockchain Technology* (submitted).

• Calvaresi, D., Dubovitskaya, A., Calbimonte, J.-P., Taveter, K., and Schumacher, M. (2018). *Multi-Agent Systems and Blockchain: Results from a Systematic Literature Review*, To Appear in Proceedings of Practical Applications of Agents and Multi-Agent Systems, ICAART 2018.

• Dubovitskaya, A., Xu, Zh., Ryu, S., Schumacher, M. and Wang, F. (2017). *Secure and Trustable Electronic Medical Records Sharing using Blockchain.* To Appear in AMIA Annual Symposium proceedings, November 4-8, 2017,Washington DC, USA

• Dubovitskaya, A., Buclin, Schumacher, M., Aberer, K. and Thoma, Y. (2017). *TUCUXI: An Intelligent System for Personalized Medicine from Individualization of Treatments to Research Databases and Back*. In Proceedings of the 8th ACMInternational Conference on Bioinformatics, Computational Biology,and Health Informatics (ACM-BCB '17). ACM, New York, NY, USA, 223-232.

• Dubovitskaya, A., Xu, Zh., Ryu, S., Schumacher, M. and Wang, F. (2017). *How Blockchain could Empower eHealth: an Application for Radiation Oncology.* In proceedings of the Third VLDB Workshop on Data Management and Analytics on Healthcare and Medicine (DMAH 2017), September 1, 2017, Munich, Germany.

• Journal paper: Dubovitskaya, A., Urovi, V., Barba, I., Aberer, K. and Schumacher, M.I. (2016). *A Multiagent System for Dynamic Data Aggregation in Medical Research.* BioMed Research International, Hindawi Publishing Corporation, 2016

• Dubovitskaya, A., Urovi, V., Vasirani, M., Aberer, K. and Schumacher, M. (2015). *A Cloud-based eHealth Architecture for Privacy Preserving Data Integration*. 30th IFIP TC-11 SEC 2015 International Information Security and Privacy Conference, IFIP, Springer Science and Business Media, 2015.

• Dubovitskaya, A., Urovi, V., Aberer, K. and Schumacher, M. (2015). *An Agent Framework for Dynamic Health Data Aggregation for Research Purposes*. IX Workshop on Agents Applied in Health Care, held in conjunction with AAMAS 2015.

• Dubovitskaya, A., Urovi, V., Vasirani, M., Aberer, K., Fuchs, A., Buclin, T., Thoma, Y., and Schumacher, M. (2014). *Privacy preserving interoperability for personalized medicine*. Swiss Medical Informatics Vol 30 (2014).

(in French)

• Dubovitskaya, A., Xu, Zh., Ryu, S., Schumacher, M. and Wang, F. (2017). *Blockchain dans la eSanté: Perspectives et une Application pour le Traitement Quotidien,* in: SwissMedical Informatics Vol 33 (2017).

(in Russian)

• Journal paper: Dubovitskaya, A., Smirnov, P. (2011). *Russian Cryptographic Algorithms in the Identity Management Systems*. Security of Information Technologies ("BIT").